HIGHER SECONDARY SECOND YEAR

COMPUTER SCIENCE

BOOK BACK QUESTION & ANSWERS

2024 - 25

Prepared By,

J. KAVITHA, B.Sc, B.Ed, M.C.A, M.Phil.,
Computer Instructor Gr - I
GHSS, S.S.KULAM
Coimbatore – 641107.
https://kavikalvi.freeweb.co.in/

Table of Contents

CHAPTER 1: Function

Choose the best answer: (1 Mark)

1. The small sections of code that are used to perform a particular task is called
   (A) Subroutines (B) Files (C) Pseudo code (D) Modules
2. Which of the following is a unit of code that is often defined within a greater code structure?
   (A) Subroutines **(B) Function** (C) Files (D) Modules
3. Which of the following is a distinct syntactic block?
   (A) Subroutines (B) Function **(C) Definition** (D) Modules
4. The variables in a function definition are called as
   (A) Subroutines (B) Function (C) Definition **(D) Parameters**
5. The values which are passed to a function definition are called
   (A) Arguments (B) Subroutines (C) Function (D) Definition
6. Which of the following are mandatory to write the type annotations in the function definition?
   (A) { } **(B) ( )** (C) [ ] (D) < >
7. Which of the following defines what an object can do?
   (A) Operating System (B) Compiler **(C) Interface** (D) Interpreter
8. Which of the following carries out the instructions defined in the interface?
   (A) Operating System (B) Compiler
   (C) Implementation (D) Interpreter
9. The functions which will give exact result when same arguments are passed are called
   (A) Impure functions (B) Partial Functions
   (C) Dynamic Functions **(D) Pure functions**
10. The functions which cause side effects to the arguments passed are called
    (A) Impure functions (B) Partial Functions
    (C) Dynamic Functions (D) Pure functions

Answer the following questions: (2 Marks)

1. **What is a subroutine?**
   - Subroutines are small sections of code that are used to perform a particular task that can be used repeatedly.
   - In Programming languages these subroutines are called as Functions.
2. **Define Function with respect to Programming language.**
   - A function is a unit of code that is often defined within a greater code structure.
   - A function works on many kinds of inputs and produces a concrete output.
3. **Write the inference you get from X:=(78).**
   - **X:=(78)** is a function definition.
   - Definitions bind values to names.
   - Hence, the value **78** bound to the name **"X"**.
4. **Differentiate interface and implementation.**

Interface	Implementation
Interface just defines what an object can do, but won't actually do it	Implementation carries out the instructions defined in the interface.

5. **Which of the following is a normal function definition and which is recursive function definition**

i) let sum x y:
  return x + y **Ans: Normal Function**

ii) let disp:
  print 'welcome' **Ans: Normal Function**

iii) let rec sum num:
  if (num!=0) then return num + sum (num-1)
  else return num **Ans: Recursive Function**

Answer the following questions: (3 Marks)

1. **Mention the characteristics of Interface.**
   - The class template specifies the interfaces to enable an object to be created and operated properly.
   - An object's attributes and behavior is controlled by sending functions to the object.
2. **Why strlen is called pure function?**
   - Each time we call the **strlen() function** with the same parameters, it always gives the same correct answer. So it is a pure function.

3. **What is the side effect of impure function? Give example.**
   - The variables used inside the function may cause side effects though the functions which are not passed with any arguments. In such cases the function is called impure function.
   - When a function depends on variables or functions outside of its definition block, you can never be sure that the function will behave the same every time it's called. **Example:** random() function.

4. **Differentiate pure and impure function.**

Pure function	Impure function
The return value of the pure functions solely depends on its arguments passed.	The return value of the impure functions does not solely depend on its arguments passed.
Pure functions will give exact result when the same arguments are passed.	Impure functions never assure you that the function will behave the same every time it's called.
They do not have any side effects.	They have side effects.
They do not modify the arguments which are passed to them.	They may modify the arguments which are passed to them.

5. **What happens if you modify a variable outside the function? Give an example.**
   - Modifying the variable outside of function causes side effect.
   - Example:

```
let y := 0
(int) inc (int)x
y := y+x;
return(y)
```

   - Here, the result of **inc()** will change every time if the value of **'y'** get changed inside the function definition.
   - Hence, the side effect of inc () function is changing the data of the external variable **'y'.**

Answer the following questions: (5Marks)

1. What are called Parameters and write a note on

(i) Parameter without Type (ii) Parameter with Type

- **Parameters** are the variables in a function definition.

(i) Parameter Without Type:

- **S**ome language compilers solve this data type problem algorithmically, if we do not specify the data types of variables in the function.

Example:

```
let rec pow a b:=
    if b=0 then 1
    else a * pow a (b-1)
```

- In the above function definition if expression can return **1** in the then branch, shows that as per the **typing** rule the entire if expression has type **int**.
- Since **'a'** is multiplied with another expression using the * operator, **'a'** must be an int.

(ii) Parameter with Type:

Example:

```
let rec pow (a: int) (b: int) : int :=
if b=0 then 1
else a * pow b (a-1)
```

- When we write the type annotations for **'a'** and **'b'** the parentheses are mandatory.
- Generally we can leave out these annotations, because it's simpler to let the compiler infer them.

2. Identify in the following program

```
let rec gcd a b :=
if b <> 0 then gcd b (a mod b) else return a
```

i) Name of the function - **gcd**

ii) Identify the statement which tells it is a recursive function - **let rec gcd a b :=**

iii) Name of the argument variable - **a, b**

iv) Statement which invoke the function recursively - **gcd b(a mod b)**

v) Statement which terminates the recursion - **return a**

3. Explain with example Pure and impure functions.

Pure functions :

- Pure functions are functions which will give exact result when the same arguments are passed.
- A function can be a pure function provided it should not have any external variable which will alter the behaviour of that variable.
- For example the mathematical function sin (0) always results **0**. This means that every time you call the function with the same arguments, you will always get the same result.

Impure functions:

- The variables used inside the function may cause side effects though the functions which are not passed with any arguments. In such cases the function is called impure function.
- When a function depends on variables or functions outside of its definition block, you can never be sure that the function will behave the same every time it's called.
- For example the mathematical function random() will give different outputs for the same function call.

4. Explain with an example interface and implementation.

- An interface is a set of action that an object can do.
- **For example**, when you press a light switch, the light goes on, you may not have cared how it splashed the light.
- Object Oriented Programming language, an Interface is a description of all functions.

The difference between interface and implementation is

Interface	Implementation
Interface just defines what an object can do, but won't actually do it	Implementation carries out the instructions defined in the interface.

- In object oriented programs classes are the interface and how the object is processed and executed is the implementation.

  Example, consider the following implementation of a function that finds the minimum of its three arguments:

```
let min 3 x y z :=
        if x < y then
                if x < z then x else z
        else
                if y < z then y else z
```

CHAPTER 2: Data Abstraction

Choose the best answer: (1 Mark)

1. Which of the following functions that build the abstract data type?
   (A) Constructors (B) Destructors (C) recursive (D) Nested
2. Which of the following functions that retrieve information from the data type?
   (A) Constructors **(B) Selectors** (C) recursive (D) Nested
3. The data structure which is a mutable ordered sequence of elements is called
   (A) Built in **(B) List** (C) Tuple (D) Derived data
4. A sequence of immutable objects is called
   (A) Built in (B) List **(C) Tuple** (D) Derived data
5. The data type whose representation is known are called
   (A) Built in datatype (B) Derived datatype
   (C) Concrete datatype (D) Abstract datatype
6. The data type whose representation is unknown are called
   (A) Built in datatype (B) Derived datatype
   (C) Concrete datatype **(D) Abstract datatype**
7. Which of the following is a compound structure?
   (A) Pair (B) Triplet (C) single (D) quadrat
8. Bundling two values together into one can be considered as
   (A) Pair (B) Triplet (C) single (D) quadrat
9. Which of the following allow to name the various parts of a multi-item object?
   (A) Tuples (B) Lists **(C) Classes** (D) quadrats
10. Which of the following is constructed by placing expressions within square brackets?
    (A) Tuples **(B) Lists** (C) Classes (D) quadrats

Answer the following questions: (2 Marks)

1. **What is abstract data type?**
   - Abstract Data type (ADT) is a type for objects or classes whose behavior is defined by a set of value and a set of operations.
2. **Differentiate constructors and selectors.**

Constructors	Selectors
Constructors are functions that build the abstract data type.	Selectors are functions that retrieve information from the data type.
Constructors create an object, bundling together different pieces of information	Selectors extract individual pieces of information from the object.

3. **What is a Pair? Give an example.**
   - To implement the data abstraction, Programming languages like Python provides a compound structure called Pair which is made up of list or Tuple.
   - Any way of bundling two values together into one can be considered as a pair. **Example:** lst[(0,10),(1,20)], Here, lst[0] = 10, lst[1] = 20
4. **What is a List? Give an example.**
   - List is constructed by placing expressions within square brackets separated by commas. List can store multiple values of any data type.

     Example: lst[10,20]
5. **What is a Tuple? Give an example.**
   - A tuple is a comma-separated sequence of values surrounded with parentheses.

     Example: Color= ('red', 'blue', 'Green')

Answer the following questions: (3 Marks)

1. **Differentiate Concrete data type and abstract data type.**

Concrete data type	Abstract data type
Concrete data types or structures (CDT's) are direct implementations of a relatively simple concept.	Abstract Data Types (ADT's) offer a high level view of a concept independent of its implementation.
A concrete data type is a data type whose representation is known.	Abstract data type the representation of a data type is unknown.

2. **Which strategy is used for program designing? Define that Strategy.**
   - A powerful strategy for designing programs is **'wishful thinking'**.
   - Wishful Thinking is the formation of beliefs and making decisions according to what might be pleasing to imagine instead of by appealing to reality.

3. **Identify Which of the following are constructors and selectors?**

(a) N1=number()	-	**Constructor**
(b) accetnum(n1)	-	**Selector**
(c) displaynum(n1)	-	**Selector**
(d) eval(a/b)	-	**Selector**
(e) x,y= makeslope (m), makeslope(n)	-	**Constructor**
(f) display()	-	**Selector**

4. **What are the different ways to access the elements of a list. Give example.**

The elements of a list can be accessed in two ways.

(1) Multiple Assignment:

- Which unpacks a list into its elements and binds each element to a different name.

Example: lst := [10, 20] x, y := lst

x will become10 and ***y*** will become 20.

(2) Element Selection Operator:

- It is expressed using square brackets.
- A second method for accessing the elements in a list is by the element selection operator.

  Example: lst[0] 10 lst[1] 20

5. **Identify Which of the following are List, Tuple and class ?**

(a) arr [1, 2, 34]	-	**List**
(b) arr (1, 2, 34)	-	**Tuple**
(c) student [rno, name, mark]	-	**Class**
(d) day= („sun‟, „mon‟, „tue‟, „wed‟)	-	**Tuple**
(e) x= [2, 5, 6.5, [5, 6], 8.2]	-	**List**
(f) employee [eno, ename, esal, eaddress]	-	**Class**

Answer the following questions: (5Marks)

1. **How will you facilitate data abstraction. Explain it with suitable example.**

To facilitate data abstraction, you will need to create constructors and selectors.

- **Constructors** are functions that build the abstract data type.
- **Selectors** are functions that retrieve information from the data type.

For example, Let's take an abstract datat ype called city.

- This city object will hold the city's name, and its latitude and longitude.

  city := makecity (name, lat, lon)
  - Here the function makecity (name, lat, lon) is the **constructor**. When it creates an object city, the values name, lat and lon are sent as parameters.
  - getname(city), getlat(city) and getlon(city) are **selector** functions that obtain information from the object city.

2. What is a List? Why List can be called as Pairs. Explain with suitable example.

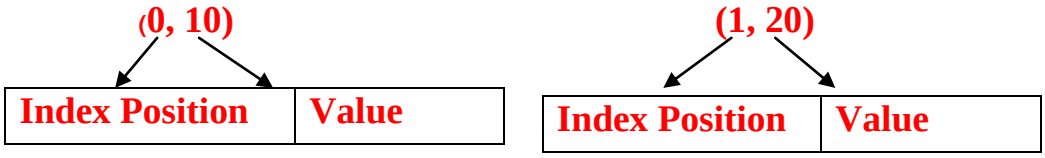
LIST:

- List is constructed by placing expressions within square brackets separated by commas. List can store multiple values of any data type.

PAIR:

- Any way of bundling two values together into one can be considered as a pair. Lists are a common method to do so. Therefore List can be called as Pairs.

Example:lst[(0,10),(1,20)] Here, lst[0] = 10, lst[1] = 20



3. How will you access the multi-item. Explain with example.

- List does not allow naming the various parts of a multi-item object.
- Instead of using a list, you can use the structure construct (In OOP languages it's called class construct) to represent multi-part objects where each part is named.

Example:

```
class Person:                 main()
person( )                     p1:=Person()
firstName := " “              firstName := " Padmashri “
id := " “                     id :="994-222-1234“
email := " “                  email="compsci@gmail.com"
```

- Same way using class you can create many objects of that type.

CHAPTER 3: Scoping

Choose the best answer: (1 Mark)

1. Which of the following refers to the visibility of variables in one part of a program to another partof the same program.
   (A) Scope (B) Memory (C) Address (D) Accessibility
2. The process of binding a variable name with an object is called
   (A) Scope **(B) Mapping** (C) late binding (D) early binding
3. Which of the following is used in programming languages to map the variable and object?
   (A):: (B) := **(C) =** (D) ==
4. Containers for mapping names of variables to objects is called
   (A) Scope (B) Mapping (C) Binding **(D) Namespaces**
5. Which scope refers to variables defined in current function?
   (A) Local Scope (B) Global scope (C) Module scope (D) Function Scope
6. The process of subdividing a computer program into separate sub-programs is called
   (A) Procedural Programming **(B) Modular programming**
   (C) Event Driven Programming (D) Object oriented Programming
7. Which of the following security technique that regulates who can use resources in a computingenvironment?
   (A) Password (B) Authentication
   (C) Access control (D) Certification
8. Which of the following members of a class can be handled only from within the class?
   (A) Public members (B) Protected members
   (C) Secured members **(D) Private members**
9. Which members are accessible from outside the class?
   (A) Public members (B) Protected members
   (C) Secured members (D) Private members
10. The members that are accessible from within the class and are also available to its subclasses iscalled
    (A) Public members **(B) Protected members**
    (C) Secured members (D) Private members

Answer the following questions: (2 Marks)

1. **What is a scope?**
   - Scope refers to the visibility of variables, parameters and functions in one part of a program to another part of the same program.
2. **Why scope should be used for variable. State the reason.**
   - The scope should be used for variables because; it limits a variable's scope to a single definition.
   - That is the variables are visible only to that part of the code.
3. **What is Mapping?**
   - The process of binding a variable name with an object is called mapping.
   - = (equal to sign) is used in programming languages to map the variable and object.
4. **What do you mean by Namespaces?**
   - Namespaces are containers for mapping names of variables to objects.
     Example: a = 5, Here the variable 'a' is mapped to the value '5'.
5. **How Python represents the private and protected Access specifiers?**
   - Python prescribes a convention of adding a prefix **__(double underscore)** results in a variable name or method becoming **private**.
     Example: self.__n2 = n2
   - Adding a prefix **_(single underscore)** to a variable name or method makes it protected. **Example: self._sal = sal**

Answer the following questions: (3 Marks)

1. **Define Local scope with an example.**
   - Local scope refers to variables defined in current function.
   - A function will always look up for a variable name in its local scope.
   - Only if it does not find it there, the outer scopes are checked.

   Example:

```
a = 10
def Inner():
    a = 20
    print(a)
Inner()
```

   Output: 20

2. **Define Global scope with an example.**
   - A variable which is declared outside of all the functions in a program is known as global variable.
   - Global variable can be accessed inside or outside of all the functions in a program.

   Example:

```
a = 10
def inner():
        a = 20
        print(a)
inner()
print(a)
```

   Output:

```
20
10
```

3. Define Enclosed scope with an example.

- A variable which is declared inside a function which contains another function definition with in it, the inner function can also access the variable of the outer function. This scope is called enclosed scope.

Example:

```
def outer():
    b = 10
    def inner():
        a = 20
        print(a)
        print(b)
    inner()
outer()
```

Output:

```
20
10
```

4. Why access control is required?

- Access control is a security technique that regulates who or what can view or use resources in a computing environment.
- It is a fundamental concept in security that minimizes risk to the object.
- In OOPS Access control is implemented through access modifiers.

5. Identify the scope of the variables in the following pseudo code and write its output

```
color:= 'Red'                - Global
mycolor():
b:='Blue'                    - Enclosed
myfavcolor():
g:='Green'                   - Local
print color, b, g
myfavcolor()
print color, b
mycolor()
print color
```

OUTPUT:

```
Red Blue Green
Red Blue
Red
```

Answer the following questions: (5 Marks)

1. Explain the types of scopes for variable or LEGB rule with example.

- Scope refers to the visibility of variables, parameters and functions in one part of a program to another part of the same program.
- The **LEGB** rule is used to decide the order in which the scopes are to be searched for scope resolution.

TYPES OF VARIABLE SCOPE:

Local(**L**)	Defined inside function/class
Enclosed(**E**)	Defined inside enclosing functions (Nested function concept)
Global(**G**)	Defined at the uppermost level
Built-in (**B**)	Reserved names in built-in functions (modules)

(i) Local scope:

- Local scope refers to variables defined in current function.
- A function will always look up for a variable name in its local scope.
- Only if it does not find it there, the outer scopes are checked.

(ii) Enclosed scope with an example.

- A variable which is declared inside a function which contains another function definition with init, the inner function can also access the variable of the outer function. This scope is called enclosed scope.

(iii) Global scope:

- A variable which is declared outside of all the functions in a program is known as global variable. Global variable can be accessed inside or outside of all the functions in a program.

(iv) Built-In-Scope:

- The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter.
- Any variable or module which is defined in the library functions of a programming language has Built-in or module scope.

Example:

```
x = 10
z = 30                              #Global
def outer():
    y = 20                                  #Enclosed
    def inner():
        x = 40                              #Local
        print(f'x is {x}')
        print(f'y is {y}')
        print(f'z is {z}')
        print(len("abc"))           #Built-in
    inner()
outer()
```

Output:

```
x is 40
y is 20
z is 30
3
```

2. **Write any Five Characteristics of Modules.**
   - Modules contain instructions, processing logic, and data.
   - Modules can be separately compiled and stored in a library.
   - Modules can be included in a program.
   - Module segments can be used by invoking a name and some parameters.
   - Module segments can be used by other modules.
3. **Write any five benefits in using modular programming.**
   - Less code to be written.
   - A single procedure can be developed for reuse.
   - Programs can be designed easily because a small team deals with only a small part of the entire code.
   - The code is stored across multiple files.
   - Code is short, simple and easy to understand.
   - Errors can easily be identified, as they are localized to a subroutine or function.
   - The same code can be used in many applications.
   - The scoping of variables can easily be controlled.

CHAPTER 4: Algorithmic Strategies

Choose the best answer: (1 Mark)

1. The word comes from the name of a Persian mathematician Abu Ja'far Mohammed ibn-i Musa alKhowarizmi is called?
   (A) Flowchart (B) Flow **(C) Algorithm** (D) Syntax
2. From the following sorting algorithms which algorithm needs the minimum number of swaps?
   (A) Bubble sort (B) Quick sort (C) Merge sort **(D) Selection sort**
3. Two main measures for the efficiency of an algorithm are
   (A) Processor and memory (B) Complexity and capacity
   (C) Time and space (D) Data and space
4. The algorithm that yields expected output for a valid input in called as
   (A) Algorithmic solution (B) Algorithmic outcomes
   (C) Algorithmic problem (D) Algorithmic coding
5. Which of the following is used to describe the worst case of an algorithm?
   (A) Big A (B) Big S (C) Big W **(D) Big O**
6. Big Ω is the reverse of
   (A) Big O (B) Big θ (C) Big A (D) Big S
7. Binary search is also called as
   (A) Linear search (B) Sequential search
   (C) Random search **(D) Half-interval search**
8. The Θ notation in asymptotic evaluation represents
   (A) Base case **(B) Average case** (C) Worst case (D) NULL case
9. If a problem can be broken into subproblems which are reused several times, the problempossesses which property?
   (A) Overlapping subproblems (B) Optimal substructure
   (C) Memoization (D) Greedy
10. In dynamic programming, the technique of storing the previously calculated values is called?
    (A) Saving value property (B) Storing value property
    (C) Memoization (D) Mapping

Answer the following questions: (2 Marks)

1. **What is an Algorithm?**
   - An algorithm is a finite set of instructions to accomplish a particular task.
   - It is a step-by-step procedure for solving a given problem**.**
2. **Write the phases of performance evaluation of an algorithm.**
   - **A Priori estimates:** This is a theoretical performance analysis of an algorithm. Efficiency of an algorithm is measured by assuming the external factors.
   - **A Posteriori testing:** This is called performance measurement. In this analysis, actual statistics like running time and required for the algorithm executions are collected.
3. **What is Insertion sort?**
   - Insertion sort is a simple sorting algorithm. It works by taking elements from the list one by one and inserting then in their correct position in to a new sorted list.
4. **What is Sorting?**
   - Sorting is a process of arranging group of items in an ascending or descending order.

     Types: Bubble Sort, Selection Sort, Insertion sort.
5. **What is searching? Write its types.**
   - A Search algorithm is the step-by-step procedure used to locate specific data among a collection of data.

     Types: Linear Search, Binary Search

Answer the following questions: (3 Marks)

1. **List the characteristics of an algorithm.**
   - Input * Output * Finiteness * Definiten
   - Effectiveness * Correctness * Simplicity * Unambiguous
   - Feasibility * Portable * Independent
2. **Discuss about Algorithmic complexity and its types.**
   - The complexity of an algorithm f(n) gives the running time and/or the storage space required by the algorithm in terms of n as the size of input data.

   Types of complexity:

   1. Time Complexity: The Time complexity of an algorithm is given by the number of steps taken by the algorithm to complete the process.

   2. Space Complexity: Space complexity of an algorithm is the amount of memory required to run to its completion.

3. **What are the factors that influence time and space complexity.**

The two main factors, which decide the efficiency of an algorithm are,

- **Time Factor** -Time is measured by counting the number of key operations like comparisons in the sorting algorithm.
- **Space Factor** - Space is measured by the maximum memory space required by the algorithm.

4. **Write a note on Asymptotic notation.**

- Asymptotic Notations are languages that use meaningful statements about time and space complexity.

  Big O - Worst-case of an algorithm.

  Big Ω - Best -case of an algorithm

  Big Θ - complexity case of an algorithm (Or) lower bound = upper bound

5. **What do you understand by Dynamic programming?**

- Dynamic programming is used when the solution to a problem can be viewed as the result of a sequence of decisions.
- The given problem will be divided into smaller overlapping sub-problems.
- An optimum solution for the given problem can be achieved by using result of smaller sub-problem.
- Dynamic algorithm uses Memorization.

Answer the following questions: (5 Marks)

1. **Explain the characteristics of an algorithm.**

Input	Zero or more quantities to be supplied.
Output	At least one quantity is produced.
Finiteness	Algorithms must terminate after finite number of steps.
Definiteness	All operations should be well defined. For example operations involving division by zero or taking square root for negative number are unacceptable.
Effectiveness	Every instruction must be carried out effectively.
Correctness	The algorithms should be error free.
Simplicity	Easy to implement.
Unambiguous	Algorithm should be clear and unambiguous. Each of its steps and their inputs/outputs should be clear and must lead to only one meaning.
Feasibility	Should be feasible with the available resources.
Portable	An algorithm should be generic, independent of any programming language or an operating system able to handle all range of inputs.
Independent	An algorithm should have step-by-step directions, which should be independent of any programming code.

2. Discuss about Linear search algorithm.

- Linear search also called sequential search is a sequential method for finding a particular value in a list.
- In this searching algorithm, list need not be ordered.

Pseudo code:

- Traverse the array using for loop.
- In every iteration, compare the target search key value with the current value of the list.
  - If the values match, display the current index and value of the array
  - If the values do not match, move on to the next array element.
- If no match is found, display the search element not found.

Example:

1. Input: values[] = {5, 34, 65, 12, 77, 35}
   target = 77
   Output: 4
2. Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}
   target = 200
   Output: -1 (not found)

3. What is Binary search? Discuss with example.

- Binary search also called half-interval search algorithm.
- It finds the position of a search element within a sorted array.

Example:

Let us assume that the **search element is 60** and we need to search the index of search element 60 using binary search.

- First, we find index of middle element by using this formula :
  mid = (low + high) / 2 , Here it is, (0 + 9) / 2 = 4.
- Compare the value stored at index 4 with target value, which is not match with search element. As the search value 60 >50.
- Now we change our search range **low to mid + 1** and find the new mid value as index 7.
- We compare the value stored at index 7 with our target value.
- Element not found because the value in index 7 is greater than search value. (80 > 60)
- Now we change our search range **low to mid - 1** and find the new mid value as index 5.
- We compare the value stored at location 5 with our search element.
- We found that it is a match.
- We can conclude that the search element 60 is found at location or index 5.
- If **no match is found** for all comparisons, then return -1.

4. Explain the Bubble sort algorithm with example.

- Bubble sort is a simple sorting algorithm; it starts at the beginning of the list of values stored in an array.
- It compares each pair of adjacent elements and swaps them if they are in the unsorted order.
- This comparison and passed to be continued until no swaps are needed, which shows the values in an array is sorted.

Pseudo code:

- Start with the first element i.e., index = 0, compare the current element with the next element of the array.
- If the current element is greater than the next element of the array, swap them.
- If the current element is less than the next or right side of the element, move to the next element.
- Go to Step 1 and repeat until end of the index is reached.

Example: Consider an array with values {15, 11, 16, 12, 14, 13}.

15 > 11 So Interchange	15	11	16	12	14	13
15 > 16 No Swapping	11	15	16	12	14	13
16 > 12 So Interchange	11	15	16	12	14	13
16 > 14 So Interchange	11	15	12	16	14	13
16 > 13 So Interchange	11	15	12	14	16	13

11	15	12	14	13	16

- The above pictorial example is for iteration-1.
- Similarly, remaining iteration can be done.
- At the end of all the iterations we will get the sorted values in an array as given below:

11	12	13	14	15	16

5. Explain the concept of Dynamic programming with suitable example.

- Dynamic programming is used when the solution to a problem can be viewed as the result of a sequence of decisions.
- Dynamic programming approach is similar to divide and conquer (i.e) the problem can be divided into smaller sub-problems.
- Results of the sub-problems can be re-used to complete the process.
- Dynamic programming approaches are used to find the solution in optimized way.

Steps to do Dynamic programming:

- The given problem will be divided into smaller overlapping sub-problems.
- An optimum solution for the given problem can be achieved by using result of smaller sub-problem.
- Dynamic algorithms uses Memoization.

Example: Fibonacci Iterative Algorithm with Dynamic Programming Approach

- Initialize f0=0, f1 =1
  step-1: Print the initial values of Fibonacci f0 and f1
  step-2: Calculate fibanocci fib ← f0 + f1
  step-3: Assign f0← f1, f1← fib
  step-4: Print the next consecutive value of fibanocci fib
  step-5: Go to step-2 and repeat until the specified number of terms generated
- For example if we generate fibonacci series up to 10 digits, the algorithm will generate the series as shown below:
  The Fibonacci series is : 0 1 1 2 3 5 8 13 21 34 55

CHAPTER 5: Python -Variables and Operators

Choose the best answer: (1 Mark)

1. Who developed Python?
   (A) Ritche **B) Guido Van Rossum** C) Bill Gates D) Sunder Pitchai
2. The Python prompt indicates that Interpreter is ready to accept instruction.
   (A) >>> B) <<< C) # D) <<
3. Which of the following shortcut is used to create new Python Program?
   (A) Ctrl + C B) Ctrl + F C) Ctrl + B **D) Ctrl + N**
4. Which of the following character is used to give comments in Python Program?
   (A) # B) & C) @ D) $
5. This symbol is used to print more than one item on a single line.
   (A) Semicolon(;) B) Dollor($) **C) comma(,)** D) Colon(:)
6. Which of the following is not a token?
   (A) Interpreter B) Identifiers C) Keyword D) Operators
7. Which of the following is not a Keyword in Python?
   (A) break B) while C) continue **D) operators**
8. Which operator is also called as Comparative operator?
   (A) Arithmetic **B) Relational** C) Logical D) Assignment
9. Which of the following is not Logical operator?
   (A) and B) or C) not **D) Assignment**
10. Which operator is also called as Conditional operator?
    (A) Ternary B) Relational C) Logical D) Assignment

Answer the following questions: (2 Marks)

1. **What are the different modes that can be used to test Python Program ?**

   In Python, programs can be written in two ways namely,

   - **Interactive mode** * **Script mode.**

2. **Write short notes on Tokens.**

   - Python breaks each logical line into a sequence of elementary lexical components known as **Tokens**.
   - The normal token types are Identifiers, Keywords, Operators, Delimiters and Literals.

3. **What are the different operators that can be used in Python ?**

   - **Operators are special symbols** which represent computations, conditional matching in programming.
   - Operators are categorized as Arithmetic, Relational, Logical, Assignment and Conditional.

4. **What is a literal? Explain the types of literal?**

   - Literal is a raw data given in a variable or constant.
   - In Python, there are various types of literals. They are,
     - **Numeric Literals -** Numeric Literals consists of digits.
     - **String literal -** string literal is a sequence of characters surrounded by quotes.
     - **Boolean literal -** A Boolean literal can have any of the two values: True or False.

5. **Write short notes on Exponent data?**

   - An Exponent data contains decimal digit part, decimal point, exponent part followed by one or more digits.

     Example:12.E04, 24.e04

Answer the following questions: (3 Marks)

1. **Write short notes on Arithmetic operator with examples.**

   - An arithmetic operator is a mathematical operator used for simple arithmetic. It takes two operands and performs a calculation on them.

   Example: Assume a=100 and b=10. Evaluate the following expressions.

Operator - Operation	Examples	Result
+ (Addition)	>>> a + b	110
- (Subtraction)	>>>a – b	90
* (Multiplication)	>>> a*b	1000
/ (Divisioin)	>>> a / b	10.0
% (Modulus)	>>> a % 30	10
** (Exponent)	>>> a ** 2	10000
// (Floor Division)	>>> a//30 (Integer Division)	3

2. What are the assignment operators that can be used in Python?

- In Python, = is a simple assignment operator to assign values to variable.
- There are various compound operators in Python like +=, -=, *=, /=, %=, **= and //= are also available.

Example: Assume x=10

Operator	Example	
=	>>> x=10 >>> b=”Computer”	
+=	>>> x+=20	# x=x+20
-=	>>> x-=5	# x=x-5
*=	>>> x*=5	# x=x*5
/=	>>> x/=2	# x=x/2
%=	>>> x%=3	# x=x%3
**=	>>> x**=2	# x=x**2
//=	>>> x//=3	

3. Explain Ternary operator with examples.

- Ternary operator is also known as **conditional operator** that evaluates something based on a condition being true or false.
- It simply allows testing a condition in a single line replacing the multiline if-else making the code compact.

Syntax: **Variable Name = [true] if [Test expression] else [false]**

Example: min = 50 if 49<50 else 70 # Output**: min = 50**

4. Write short notes on Escape sequences with examples.

- In Python strings, the backslash "\" is a special character, also called the "**escape**" character.
- It is used in representing certain whitespace characters:
  - "**\t**" is a tab
  - "**\n**" is a newline
  - "**\r**" is a carriage return.

For example to print the message "It's raining", the Python command is

```
>>> print ("It\'s rainning")
        It's rainning
```

5. What are string literals? Explain.

- In Python a string literal is a **sequence of characters** surrounded by **quotes**.
- Python supports **single, double and triple quotes** for a string.
- A character literal is a **single character** surrounded by **single or double quotes**.
- The value with **triple-quote** "' '" is used to give **multi-line** string literal.
- **Example:** strings = "This is Python"
  char = ‘C’

Answer the following questions: (5 Marks)

1. **Describe in detail the procedure Script mode programming.**

- A script is a text file containing the Python statements.
- Once the Python Scripts is created, they are reusable; it can be executed again and again without retyping.

(i) Creating Scripts in Python:

- Choose **File→ New File** or press **Ctrl + N** in Python shell window.
- An **untitled** blank script text editor will be displayed on screen.
- Type the code in Script editor

(ii) Saving Python Script:

- Choose **File→ Save** or Press **Ctrl + S**
- Now, **Save As** dialog box appears on the screen.
- Type the file name with extension **.py** in **File Name** box.
- Then click **Save** button to save your Python script.

(iii) Executing Python Script:

- Choose **Run → Run Module** or Press **F5**
- If your code has any error, it will be shown in red color in the IDLE window, and Python describes the type of error occurred.
- To correct the errors, go back to Script editor, make corrections, save the file and execute it again.
- For all error free code, the output will appear in the IDLE window of Python.

2. **Explain input() and print() functions with examples.**

1) input() function:

- In Python, **input( )** function is used to accept data as input at run time.

The syntax for **input()** function is,

Variable = input("prompt string")

- **"Prompt string"** in the syntax is a message to the user, to know what input can be given.
- The **input( )** takes typed data from the keyboard and stores in the given variable.
- If prompt string is not given in **input( )**, the user will not know what is to be typed as input.

Example: >>> city=input ("Enter Your City: ")

Output: Enter Your City: Madurai

2) print() function:

- In Python, the **print()** function is used to display result on the screen.

Syntax for print(): **print("String")**
print(variable)

- The print () displays an entire statement which is specified within print().
- **Comma (,)** is used as a separator in **print ( )** to print more than one item.

Example: >>> print ("Welcome to Python Programming")

Output: Welcome to Python Programming

3. Discuss in detail about Tokens in Python.

- Python breaks each logical line into a sequence of elementary lexical components known as **Tokens**.
- The normal token types are Identifiers, Keywords, Operators, Delimiters and Literals.

1) Identifiers:

- An Identifier is a name used to identify a variable, function, class, module or object.
- An identifier must start with an alphabet (A..Z or a..z) or underscore (_).
- Identifiers may contain digits (0 .. 9)
- Python identifiers are case sensitive i.e. uppercase and lowercase letters are distinct.
- Identifiers must not be a **python** keyword.

Example: Sum, total_marks, regno, num1

2) Keywords:

- Keywords are special words used by Python interpreter to recognize the structure of program.
- Keywords have **specific meaning for interpreter**; they cannot be used for any other purpose.

Python Keywords: false, class, If, elif, else, pass, break etc.

3) Operators:

- **Operators are special symbols** which represent computations, conditional matching in programming.
- Operators are categorized as Arithmetic, Relational, Logical, Assignment and Conditional.

Example:
```
a=100
b=10
print ("The Sum = ",a+b)
```
Output: The Sum = 110

4) Delimiters:

- Python uses the symbols and symbol combinations as delimiters in expressions, lists, dictionaries and strings.

Following are the delimiters: (,), {, }, [,], :, ;, +=, *= ....

5) Literals:

- Literal is a raw data given in a variable or constant.
- In Python, there are various types of literals. They are,
  - **Numeric Literals** consists of digits and are immutable.
  - **String literal** is a sequence of characters surrounded by quotes.
  - **Boolean literal** can have any of the two values: True or False.

CHAPTER 6: Control Structures

Choose the best answer: (1 Mark)

1. How many important control structures are there in Python?
   (A) 3 B) 4 C) 5 D) 6
2. elif can be considered to be abbreviation of
   (A) nested if B) if..else **C) else if** D) if..elif
3. What plays a vital role in Python programming?
   (A) Statements B) Control C) Structure **D) Indentation**
4. Which statement is generally used as a placeholder?
   (A) continue B) break **C) pass** D) goto
5. The condition in the if statement should be in the form of
   (A) Arithmetic or Relational expression B) Arithmetic or Logical expression
   C) Relational or Logical expression D) Arithmetic
6. Which is the most comfortable loop?
   (A) do..while B) while **C) for** D) if..elif
7. What is the output of the following snippet?

```
i=1
while True:
    if i%3 ==0:
    break
    print(i,end='')
    i +=1
```

   (A) 12 B) 123 C) 1234 D) 124
8. What is the output of the following snippet?

```
T=1
while T:
print(True)
 break
```

   (A) False **B) True** C) 0 D) 1
9. Which amongst this is not a jump statement?
   (A) for B) pass C) continue D) break
10. Which punctuation should be used in the blank?

```
if <condition>_
    statements-block 1
else:
    statements-block 2
```

   (A) ; **B) :** C) :: D) !

Answer the following questions: (2 Marks)

1. **List the control structures in Python**.

   Three important control structures are,

   - Sequential * Alternative or Branching * Iterative or Looping

2. **Write note on break statement.**
   - The **break** statement terminates the loop containing it.
   - Control of the program flows to the statement immediately after the body of the loop.
3. **Write is the syntax of if..else statement.**

   Syntax:

```
if <condition>:
        statements-block 1
else:
        statements-block 2
```

4. **Define control structure.**
   - A program statement that causes a jump of control from one part of the program to another is called control structure or control statement.
5. **Write note on range () in loop.**
   - range() generates a list of values starting from start till stop-1 in for loop.

   The syntax of range() is as follows:

   range (start,stop,[step])
   - Where, **start**– refers to the initial value
   - **stop**– refers to the final value
   - **step**– refers to increment value, this is optional part.

Answer the following questions: (3 Marks)

1. **Write a program to display**

   A

   A B

   A B C

   A B C D

   A B C D E

Coding:

```
a=['A','B','C','D','E']
for i in range(0,6):
        for j in range(0,i):
                print(a[j],end=" ")
        else:
                print()
```

2. Write note on if..else structure.

- The **if .. else** statement provides control to check the true block as well as the false block.
- **if..else** statement thus provides two possibilities and the condition determines which BLOCK is to be executed.

Syntax:
```
if <condition>:
        statements-block 1
else:
        statements-block 2
```

3. Using if..else..elif statement write a suitable program to display largest of 3 numbers.

Coding:
```
a=int(input("Enter Number 1:"))
b=int(input("Enter Number 2:"))
c=int(input("Enter Number 3:"))
if a>b and a>c:
        print(a,"is biggest")
elif b>a and b>c:
        print(b,"is biggest")
else:
        print(c,"is biggest")
```

OUTPUT:
```
Enter Number 1:50
Enter Number 2:14
Enter Number 3:25
50 is biggest
```

4. Write the syntax of while loop.

Syntax:
```
while <condition>:
        statements block 1
[else:
        statements block2]
```

5. List the differences between break and continue statements.

break	continue
The **break** statement terminates the loop containing it.	The**Continue** statement is used to skip the remaining part of a loop
Control of the program flows to the statement immediately after the body of the loop. **Syntax:** break	Control of the program flows start with next iteration. **Syntax:** continue

Answer the following questions: (5 Marks)

1. Write a detail note on for loop.

- **for** loop is the most comfortable loop. It is also an entry check loop.
- The condition is checked in the beginning and the body of the loop (statements-block 1) is executed if it is only True otherwise the loop is not executed.

Syntax:

```
for counter_variable in sequence:
        statements-block 1
[else:          # optional block
        statements-block 2]
```

- The counter_variable is the control variable.
- The sequence refers to the initial, final and increment value.
- **for** loop uses the range()function in the sequence to specify the initial, final and increment values.
- range() generates a list of values starting from start till stop-1 in for loop.

The syntax of range() is as follows:

```
range (start,stop,[step])
```

- Where, **start**– refers to the initial value
- **stop**– refers to the final value
- **step**– refers to increment value, this is optional part.

Example:

```
for i in range(2,10,2):
        print (i,end=' ')
```

Output: 2 4 6 8

2. Write a detail note on if..else..elif statement with suitable example.

- When we need to construct a chain of **if** statement(s) then **'elif'** clause can be used instead of **'else'.**

Syntax:

```
if <condition-1>:
        statements-block 1
elif <condition-2>:
        statements-block 2
else:
        statements-block n
```

- In the syntax of **if..elif..else** mentioned above, condition-1 is tested if it is true then statements-block1 is executed.
- Otherwise the control checks condition-2, if it is true statements-block2 is executed and even if it fails statements-block n mentioned in **else** part is executed.

Example:

```
m1=int (input("Enter mark in first subject : "))
m2=int (input("Enter mark in second subject : "))
avg= (m1+m2)/2
if avg>=80:   print ("Grade : A")
elif avg>=70 and avg<80:   print ("Grade : B")
elif avg>=60 and avg<70:   print ("Grade : C")
elif avg>=50 and avg<60:   print ("Grade : D")
else:   print ("Grade : E")
```

Output:

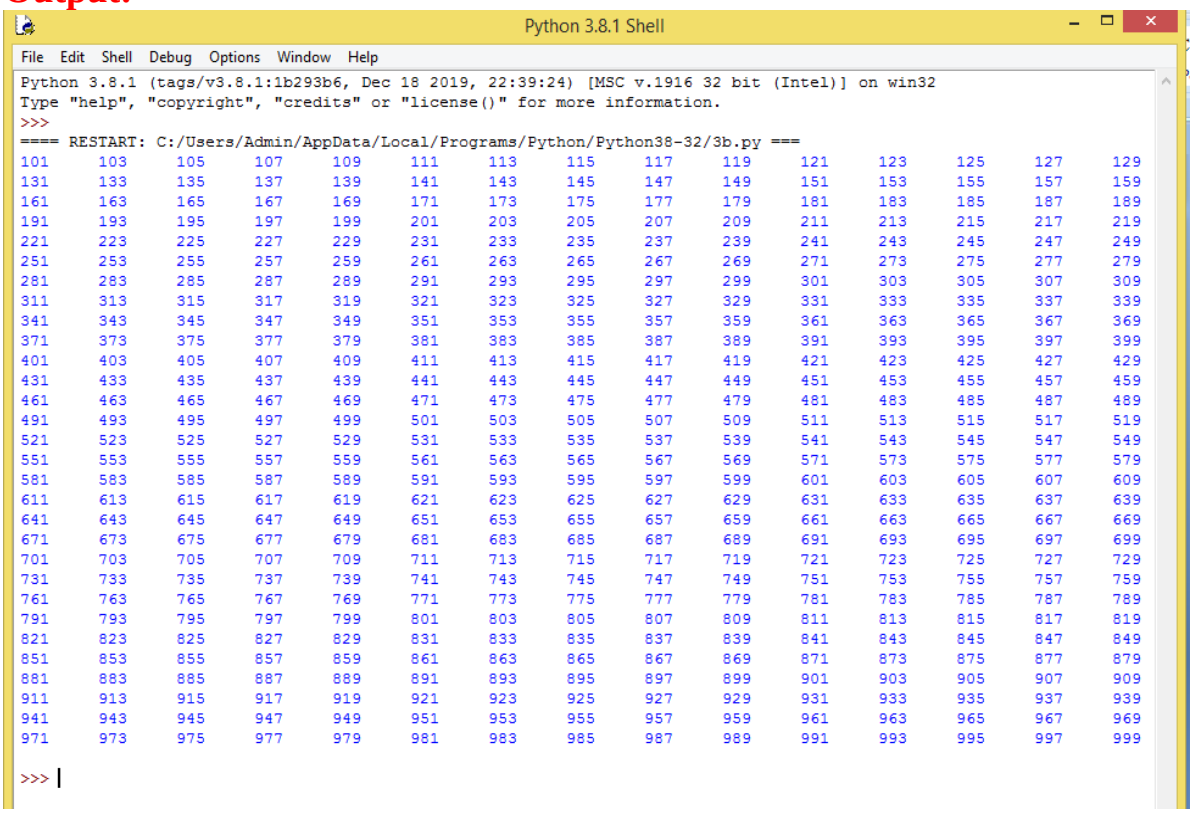
```
Enter mark in first subject : 34
Enter mark in second subject : 78
Grade : D
```

3. Write a program to display all 3 digit odd numbers.

Coding:

```
for i in range(101,1000,2):
    print(i,end='\t')
```

Output:



4. Write a program to display multiplication table for a given number.

Coding:

```
num=int(input("Display Multiplication Table of "))
for i in range(1,11):
    print(i, 'x' ,num, '=' , num*i)
```

Output:

```
Display Multiplication Table of 7
1 x 7 = 7
2 x 7 = 14
3 x 7 = 21
4 x 7 = 28
5 x 7 = 35
6 x 7 = 42
7 x 7 = 49
8 x 7 = 56
9 x 7 = 63
10 x 7 = 70
```

CHAPTER 7: Python functions

Choose the best answer: (1 Mark)

1. A named blocks of code that are designed to do one specific job is called as
   (A) Loop (B) Branching **(C) Function** (D) Block
2. A Function which calls itself is called as
   (A) Built-in **(B) Recursion** (C) Lambda (D) return
3. Which function is called anonymous un-named function
   (A) Lambda (B) Recursion (C) Function (D) define
4. Which of the following keyword is used to begin the function block?
   (A) define (B) for (C) finally **(D) def**
5. Which of the following keyword is used to exit a function block?
   (A) define **(B) return** (C) finally (D) def
6. While defining a function which of the following symbol is used.
   (A) ; (semicolon) (B) . (dot) **(C) : (colon)** (D) $ (dollar)
7. In which arguments the correct positional order is passed to a function?
   (A) Required (B) Keyword (C) Default (D) Variable-length
8. Read the following statement and choose the correct statement(s).
   (I) In Python, you don't have to mention the specific data types while defining function.
   (II) Python keywords can be used as function name.
   (A) I is correct and II is wrong (B) Both are correct
   (C) I is wrong and II is correct (D) Both are wrong
9. Pick the correct one to execute the given statement successfully.if___:
   print(x, " is a leap year")
   (A) x%2=0 **(B) x%4==0** (C) x/4=0 (D) x%4=0
10. Which of the following keyword is used to define the function test python(): ?
    (A) define (B) pass **(C) def** (D) while

Answer the following questions: (2 Marks)

1. **What is function?**
   - Functions are named blocks of code that are designed to do one specific job.
2. **Write the different types of function.**
   - User-defined Functions
   - Built-in Functions
   - Lambda Functions
   - Recursion Functions
3. **What are the main advantages of function?**
   - It avoids repetition and makes high degree of code reusing.
   - It provides better modularity for your application.
4. **What is meant by scope of variable? Mention its types.**
   - Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can use it.

   Types: 1) local scope 2) global scope.
5. **Define global scope.**
   - A variable, with global scope can be used anywhere in the program.
   - It can be created by defining a variable outside the scope of any function.
6. **What is base condition in recursive function.**
   - A recursive function calls itself.
   - The condition that is applied in any recursive function is known as base condition.
   - A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.
7. **How to set the limit for recursive function? Give an example.**
   - Python stops calling recursive function after 1000 calls by default.
   - So, It also allows you to change the limit using sys.setrecursionlimit (limit_value).

   Example:

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact(n-1)
print(fact (2000))
```

Answer the following questions: (3 Marks)

1. **Write the rules of local variable.**
   - A variable with local scope can be accessed only within the block that it is created in.
   - When a variable is created inside the function, the variable becomes local to it.
   - A local variable only exists while the function is executing.
   - The formal arguments are also local to function.
2. **Write the basic rules for global keyword in python.**
   - When we define a variable outside a function, it's global by default. We don't have to use global keyword.
   - We use global keyword to read and write a global variable inside a function.
   - Use of global keyword outside a function has no effect.
3. **What happens when we modify global variable inside the function?**
   - If we modify the global variable inside the function, it will show Unbound Local Error.

   Example : Modifying Global Variable From Inside the Function

```
c = 1                          # global variable
def add():
c = c + 2                      # increment c by 2
print(c)
add()
```

   Output: UnboundLocal Error: local variable 'c' referenced before assignment

4. **Differentiate ceil() and floor() function?**

ceil()	floor()
Returns the smallest integer greater than or equal to x.	Returns the largest integer less than or equal to x.
Syntax: math.ceil (x)	Syntax: math.floor (x)
Ex: >>>import math >>>print(math.ceil(26.7)) Output: 27 >>>Print(math.ceil(-26.7)) Output: -26	Ex: >>>import math >>>print(math.floor(26.7)) Output: 26 >>>Print(math.floor(-26.7)) Output: -27

5. **Write a Python code to check whether a given year is leap year or not.**

Coding:

```
n=int(input("Enter the year"))
 if(n%4==0):
        print (n, "is a Leap Year")
else:
        print (n, "is not a Leap Year")
```

Output:

```
Enter the year 2012
2012 is a Leap Year
```

6. What is composition in functions?

- The value returned by a function may be used as an argument for another function in a nested manner. This is called **composition**.

For example, if we wish to take a numeric value as a input from the user, we take the input string from the user using the function **input()** and apply **eval()**function to evaluate its value.

```
>>> n1 = eval(input("Enter an Airthmetic Expression:"))
Enter an Airthmetic Expression:12.0+13.0*2
>>> n1
38.0
```

7. How recursive function works?

- Recursive function is called by some external code.
- If the base condition is met then the program gives meaningful output and exits.
- Otherwise, function does some required processing and then calls itself to continue recursion.

Example:

```
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact (n-1)
print (fact (0))
print (fact (5))
```

Output:

```
1
120
```

8. What are the points to be noted while defining a function?

- Function blocks begin with the keyword **"def"** followed by function name and parenthesis ().
- Any input parameters should be placed within these parentheses.
- The code block always comes after a colon (:) and is indented.
- The statement **"return [expression]"** exits a function, and it is optional. A**"return"** with no arguments is the same as return None.

Syntax:

```
def <function_name ([parameter1, parameter2…] )> :
    <Block of Statements>
    return <expression / None>
```

Answer the following questions: (5 Marks)

1. Explain the different types of function with an example.

- Functions are named blocks of code that are designed to do one specific job.

Types of Functions:

User defined Function:

- Functions defined by the users themselves are called user defined function.

Example:
```
def hello():
        print ("hello - Python")
        return
```
Output: hello – Python

Built-in Function:

- Built-in functions are Functions that are inbuilt with in Python.

Example: print(), echo() are some built-in function.

Lambda Function:

- In Python, anonymous function is a function that is defined without a name.
- While normal functions are defined using the **def** keyword, in Python anonymous functions are defined using the **lambda** keyword.
- Hence, anonymous functions are also called as **lambda** functions.

Example:
```
sum = lambda arg1, arg2: arg1 + arg2
        print ('The Sum is :', sum(30,40))
        print ('The Sum is :', sum(-30,40))
```
Output:
```
The Sum is : 70
The Sum is : 10
```
Recursion Function:

- Functions that calls itself is known as recursive.

Example:
```
def fact(n):
        if n == 0:
                return 1
        else:
                return n * fact (n-1)
print (fact (0))
print (fact (5))
```
Output:
```
1
120
```

2. Explain the scope of variables with an example.

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can use it.
- There are two types of scopes: **local scope** and **global scope**.

Local Scope:

- A variable declared inside the function's body or in the local scope is known as local variable.

Rules of local variable:

- A variable with local scope can be accessed only within the function/block that it is created in.
- When a variable is created inside the function/block, the variable becomes local to it.
- A local variable only exists while the function is executing. The formal arguments are also local to function.

Example:

```
def loc():
        y=0                 # local scope
        print(y)
loc()
```

Output: 0

Global Scope:

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

Rules of global Keyword:

- When we define a variable outside a function, it's global by default. We don't have to use global keyword.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect

Example:

```
c = 1                   # global variable
def add():
        print(c)
add()
```

Output: **1**

3. Explain the following built-in functions.

(a) id() (b) chr() (c) round() (d) type() (e) pow()

Function	Description	Example
id ( )	Return the address of the object in memory.	x=15 print ('address of x is :',id (x)) **Output:** address of x is : 1357486752
chr ( )	Returns the Unicode character for the given ASCII value.	c=65 print(chr(c)) **Output:** A
round ( )	Returns the nearest integer to its input.	x= 17.9 print (round (x)) **Output:** 18
type ( )	Returns the type of object for the given single object.	x= 15.2 print (type (x)) **Output:** <class 'float'>
pow ( )	Returns the computation of a,b i.e. (a**b) a raised to the power of b.	a= 5 b= 2 print (pow (a,b)) **Output:** 25

4. Write a Python code to find the L.C.M. of two numbers.

Coding:

```
def lcm(x,y):
    if x>y:
        greater = x
    else:
        greater = y
    while(True):
        if((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1
    return lcm
a = int(input("Enter first number:"))
b = int(input("Enter second number:"))
print("LCM is", lcm(a,b))
```

OUTPUT:
```
Enter first number: 2
Enter second number: 3
LCM is: 6
```

5. **Explain recursive function with an example.**

- Functions that calls itself is known as recursive.
- Recursion works like loop but sometimes it makes more sense to use recursion than loop.
- Imagine a process would iterate indefinitely if not stopped by some condition is known as infinite iteration.
- The condition that is applied in any recursive function is known as base condition.
- A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

Overview of how recursive function works:

- Recursive function is called by some external code.
- If the base condition is met then the program gives meaningful output and exits.
- Otherwise, function does some required processing and then calls itself to continue recursion.

Example:

```
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact (n-1)
    print (fact (0))
    print (fact (5))
```

Output:

```
1
120
```

CHAPTER 8: Strings and String manipulation

Choose the best answer: (1 Mark)

1. Which of the following is the output of the following python code?
```
str1="TamilNadu"
print(str1[::-1])
```
(A) Tamilnadu (B) Tmlau (C) udanlimaT **(D) udaNlimaT**

2. What will be the output of the following code?
```
str1 = "Chennai Schools"
str1[7] = "-"
```
(A) Chennai-Schools (B) Chenna-School
(C) Type error (D) Chennai

3. Which of the following operator is used for concatenation?
(A) + (B) & (C) * (D) =

4. Defining strings within triple quotes allows creating:
(A) Single line Strings **(B) Multiline Strings**
(C) Double line Strings (D) Multiple Strings

5. Strings in python:
(A) Changeable (B) Mutable **(C) Immutable** (D) flexible

6. Which of the following is the slicing operator?
(A) { } **(B) [ ]** (C) < > (D) ( )

7. What is stride?
(A) index value of slide operation (B) first argument of slice operation
(C) second argument of slice operation **(D) third argument of slice operation**

8. Which of the following formatting character is used to print exponential notation in upper case?
(A) %e **(B) %E** (C) %g (D) %n

9. Which of the following is used as placeholders or replacement fields which get replaced alongwith format( ) function?
(A) { } (B) < > (C) ++ (D) ^^

10. The subscript of a string may be:
(A) Positive (B) Negative
(C) Both (A) and (B) **(D) Either (A) or (B)**

Answer the following questions: (2 Marks)

1. **What is String?**
   - String is a data type in python, used to handle array of characters.
   - String is a sequence of characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.
2. **Do you modify a string in Python?**
   - Strings in python are immutable. That means, once you define a string modifications or deletion is not allowed.
   - However, we can replace the existing string entirely with the new string.
3. **How will you delete a string in Python?**
   - Python will not allow deleting a particular character in a string.
   - Whereas you can remove entire string variable using **del** command.
4. **What will be the output of the following python code?**

```
str1 = “School”
print(str1*3)
```

   Output: School School School
5. **What is slicing?**
   - Slice is a substring of a main string.
   - A substring can be taken from the original string by using [ ] slicing operator and index or subscript values.
   - Using slice operator, we have to slice one or more substrings from a main string.

Answer the following questions: (3 Marks)

1. **Write a Python program to display the given pattern**

```
C O M P U T E R
C O M P U T E
C O M P U T
C O M P U
C O M P
C O M
C O
C
```

Coding:

```
str="COMPUTER"
index=len(str)
for i in str:
        print(str[:index])
        index-=1
```

2. Write a short about the followings with suitable example:
(a) capitalize( ) (b) swapcase( )

Function	Purpose	Example

capitalize( )	Used to capitalize the first character of the string	>>> city="chennai"
		>>> print(city.capitalize())
		Output: Chennai

swapcase( )	It will change case of every character to its opposite case vice-versa.	>>> str1="tAmiL NaDu" >>>print(str1.swapcase()) **Output:TaMIl nAdU**

3. What will be the output of the given python program?

```
str1 = "welcome"
str2 = "to school"
str3=str1[:2]+str2[len(str2)-2:]
print(str3)
```

Output: weol

4. What is the use of format( )? Give an example.

- The **format( )**function used with strings is very powerful function used for formatting strings.
- The curly braces { } are used as placeholders or replacement fields which get replaced along with format( ) function.

Example:

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
```

Output:
```
Number 1: 34
Number 2: 54
The sum of 34 and 54 is 88
```

5. Write a note about count( ) function in python.

- Returns the number of substrings occurs within the given range.
- Remember that substring may be a single character.

Syntax: count(str, beg, end)

- Range (beg and end) arguments are optional.

Example:
```
>>> str1="Raja Raja Chozhan"
>>> print(str1.count('Raja'))
```

Output: 2

Answer the following questions: (5 Marks)

1. Explain about string operators in python with suitable example.

Python provides the following string operators to manipulate string.

Concatenation (+):

- Joining of two or more strings using plus **(+) operator** is called as **Concatenation**.

Example: >>> "welcome" + "Python"

Output: **'welcomePython'**

Append (+ =):

- Adding more strings at the end of an existing string using **operator +=** is known as **append.**

Example:
```
>>> str1="Welcome to "
>>> str1+="Learn Python"
>>> print (str1)
```
Output: **Welcome to Learn Python**

Repeating (*):

- The multiplication operator (*) is used to display a string in multiple number of times.

Example:
```
>>> str1="Welcome "
>>> print (str1*4)
```
Output: Welcome Welcome Welcome Welcome

String slicing([ ]):

- Slice is a substring of a main string.
- A substring can be taken from the original string by using **[ ] slicing operator** and index values.
- Using slice operator, we have to slice one or more substrings from a main string.

Example:
```
>>> str1="THIRUKKURAL"
>>> print (str1[0])
```
Output: T

Stride when slicing string:

- When the slicing operation, we can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string.
- The default value of stride is 1

Example:
```
>>> str1 = "Welcome to learn Python"
>>> print (str1[10:16])
>>> print(str1[::-2])
```
Output: Learn
nhy re teolW

CHAPTER 9: Lists, Tuples, Sets and Dictionary

Choose the best answer: (1 Mark)

1. Pick odd one in connection with collection data type
   (A) List (B) Tuple (C) Dictionary **(D) Loop**
2. Let list1=[2,4,6,8,10], then print(List1[-2]) will result in
   (A) 10 **(B) 8** (C) 4 (D) 6
3. Which of the following function is used to count the number of elements in a list?
   (A) count() (B) find() **(C) len()** (D) index()
4. If List=[10,20,30,40,50] then List[2]=35 will result
   (A) [35,10,20,30,40,50] (B) [10,20,30,40,50,35]
   (C) [10,20,35,40,50] (D) [10,35,30,40,50]
5. If List=[17,23,41,10] then List.append(32) will result
   (A) [32,17,23,41,10] **(B) [17,23,41,10,32]**
   (C) [10,17,23,32,41] (D) [41,32,23,17,10]
6. Which of the following Python function can be used to add more than one element within anexisting list?
   (A) append() (B) append_more() **(C) extend()** (D) more()
7. What will be the result of the following Python code? S=[x**2 for x in range(5)]
   print(S)
   (A) [0,1,2,4,5] **(B) [0,1,4,9,16]** (C) [0,1,4,9,16,25] (D) [1,4,9,16,25]
8. What is the use of type() function in python?
   (A) To create a Tuple
   (B) To know the type of an element in tuple.
   (C) To know the data type of python object.
   (D) To create a list.
9. Which of the following statement is not correct?
   (A) A list is mutable (B) A tuple is immutable.
   (C) The append() function is used to add an element.
   (D) The extend() function is used in tuple to add elements in a list.
10. Let setA={3,6,9}, setB={1,3,9}. What will be the result of the following snippet?print(setA|setB)
    (A) {3,6,9,1,3,9} (B) {3,9} (C) {1} **(D) {1,3,6,9}**
11. Which of the following set operation includes all the elements that are in two sets but not the onethat are common to two sets?
    (A) Symmetric difference (B) Difference (C) Intersection (D) Union
12. The keys in Python, dictionary is specified by
    (A) = (B) ; (C)+ **(D) :**

Answer the following questions: (2 Marks)

1. **What is List in Python?**
   - A list is an ordered collection of values enclosed within square brackets [ ] also known as a 'sequence data type'.
   - Each value of a list is called as element.
   - Elements can be a numbers, characters, strings and even the nested lists.

   Syntax: Variable = [element-1, element-2, element-3 …… element-n]
2. **How will you access the list elements in reverse order?**
   - A negative index can be used to access an element in reverse order.
3. **What will be the value of x in following python code?**

   List1=[2,4,6[1,3,5]]

   x=len(List1)

   Output: 4
4. **Differentiate del with remove() function of List.**

del	remove( )
del statement is used to delete known elements	**remove( )**function is used to delete elements of a list if its index is unknown.

5. **Write the syntax of creating a Tuple with n number of elements.**

   Syntax: **Tuple_Name = (E1, E2, E2 ……. En)** Or

   Tuple_Name = E1, E2, E3 ….. En
6. **What is set in Python?**
   - In python, a set is another type of collection data type.
   - A Set is a mutable and an unordered collection of elements without duplicates or repeated element.

Answer the following questions: (3 Marks)

1. **What are the difference between list and Tuples?**
   - The elements of a list are changeable (mutable) whereas the elements of a tuple are unchangeable(immutable), this is the key difference between tuples and list.
   - The elements of a list are enclosed within square brackets. But, the elements of a tuple are enclosed by parenthesis.
   - Iterating tuples is faster than list.
2. **Write a shot note about sort().**
   - Sort() function sorts the element in list.

   Syntax : List.sort(reverse=True/False, key=myFunc)
   - If reverse is set as True, list sorting is in descending order. Ascending is default.

3. **What will be the output of the following code?**

```
list = [2**x for x in range(5)]
print(list)
```

Output: [1, 2, 4, 8, 16]

4. **Explain the difference between del and clear() in dictionary with an example.**

del	clear( )
The **del** statement is used to delete known elements	The function **clear( )** is used to delete all the elements in list
The del statement can also be used to delete entire list.	It deletes only the elements and retains the list.

Ex:
>>> MySubjects = ['Tamil', 'English']
>>> del MySubjects[1]
>>> print (MySubjects)
Output: ['Tamil']
	Ex:
	>>> MySubjects = ['Tamil', 'English']
	>>> MySubjects.clear()
	>>> print (MySubjects)
	Output: [ ]

5. **List out the set operations supported by python.**

- **Union (|):** It includes all elements from two or more sets.
- **Intersection (&):** It includes the common elements in two sets.
- **Difference (-):** It includes all elements that are in first set but not in the second set.
- **Symmetric difference (^):** It includes all the elements that are in two sets but not the one that are common to two sets.

6. **What are the difference between List and Dictionary?**

List	Dictionary
List is an ordered set of elements.	Dictionary is a data structure that is used for matching one element (Key) with another (Value).
The index values can be used to access a particular element.	In dictionary key represents index.
Lists are used to look up a value	dictionary is used to take one value and look up another value.

Answer the following questions: (5 Marks)

1. **What the different ways to insert an element in a list. Explain with suitable example.**

- **append():** The **append( )**function is used to add a single element at the end of a list.

Syntax: List.append (element to be added)

Example:
```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
```

Output: [34, 45, 48, 90]

- **extend():** The **extend( )**function is used to add more than one element to an existing list.

Syntax: List.extend ([elements to be added])

Example:
```
>>> Mylist=[34, 45, 48]
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
```

Output: [34, 45, 48, 90, 71, 32, 29]

- **insert():** The **insert( )** function is used to insert an element at any position of a list.

Syntax: List.insert (position index, element)

Example:
```
>>>MyList=[34,98,47,'Kannan','Sankar', 'Lenin', 'Sreenivasan' ]
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
```

Output: [34,98,47,'Ramakrishnan','Kannan',’Sankar', 'Lenin', 'Sreenivasan']

2. **What is the purpose of range()? Explain with an example.**

- Using range( ) function, we can create list with series of values.
- The range( ) function has three arguments.

Syntax: range (start value, end value, step value)

- **start value** – beginning value of series.
- **end value** – upper limit of series.
- **step value** – It is an optional argument, which is used to generate different interval of values.

Example :
```
for x in range (2, 11,2):
        print(x, end=’ ‘)
```

Output: 2 4 6 8 10

3. **What is nested tuple? Explain with an example.**

- In Python, a tuple can be defined inside another tuple called Nested tuple.
- In a nested tuple, each tuple is considered as an element.
- The for loop will be useful to access all the elements in a nested tuple.

Example:
```
Toppers = (("Vinothini", "XII-F", 98.7), ("Tharani", "XII-F", 95.3))
for i in Toppers:
        print(i)
```

Output:
```
('Vinodini', 'XII-F', 98.7)
('Tharani', 'XII-F', 95.3)
```

4. Explain the different set operations supported by python with suitable example.

- **Union(|):** It includes all elements from two or more sets. The **operator |** is used to union of two sets.

Example:
```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```
Output: {2, 4, 6, 8, 'A', 'D', 'C', 'B'}

- **Intersection(&):** It includes the common elements in two sets. The **operator &** is used to intersect two sets in python.

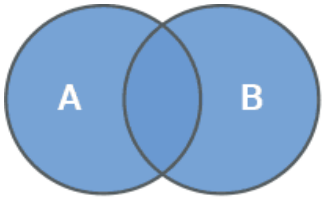
Example:
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```
Output: {'A', 'D'}

- **Difference(-):** It includes all elements that are in first set but not in the second set. The minus **(-) operator** is used to difference set operation in python.

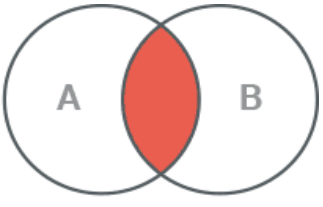
Example:
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```
Output: {2, 4}

- **Symmetric difference(^):** It includes all the elements that are in two sets but not the one that are common to two sets. The caret **(^) operator** is used to symmetric difference set operation in python.

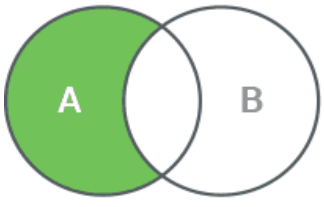
Example:
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A ^ set_B)
```
Output: {2, 4, 'B', 'C'}



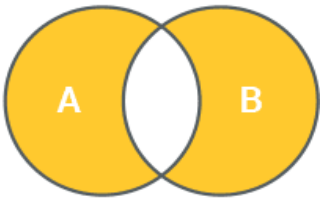
Union



Intersection



Difference



Symmetric Difference

CHAPTER 10: Python Classes and objects

Choose the best answer: (1 Mark)

1. Which of the following are the key features of an Object Oriented Programming language?
   (A) Constructor and Classes (B) Constructor and Object
   (C) Classes and Objects (D) Constructor and Destructor
2. Functions defined inside a class:
   (A) Functions (B) Module **(C) Methods** (D) section
3. Class members are accessed through which operator?
   (A) & **(B) .** (C) # (D) %
4. Which of the following method is automatically executed when an object is created?
   (A) __object__( ) (B)__del__( ) (C)__func__( ) **(D)__init__( )**
5. A private class variable is prefixed with
   (A) __ (B) && (C) ## (D) **
6. Which of the following method is used as destructor?
   (A)__init__( ) (B)__dest__( ) (C)__rem__( ) **(D)__del__( )**
7. Which of the following class declaration is correct?
   (A) class class_name (B) class class_name<>
   (C) class class_name: (D) class class_name[ ]
8. Which of the following is the output of the following program?

```
class Student:
    def__init_(self, name):
    self.name=name
    print (self.name)
    S=Student(Tamil)
```

   (A) Error **(B) Tamil** (C) name (D) self
9. Which of the following is the private class variable?
   (A) __num (B) ##num (C) $$num (D) &&num
10. The process of creating an object is called as:
    (A) Constructor (B) Destructor (C) Initialize **(D) Instantiation**

Answer the following questions: (2 Marks)

1. **What is class?**
   - Class is the main building block in Python.
   - Class is a template for the object.
2. **What is instantiation?**
   - The process of creating object is called as "Class Instantiation".
3. **What is the output of the following program?**

```
class Sample:
    __num=10
    def disp(self):
        print(self.__num)
S=Sample()
S.disp()
print(S.__num)
```

Ootput:

```
>>>10        line 7, in <module>
print(S.__num)
AttributeError: 'Sample' object has no attribute '__num'
>>>
```

4. **How will you create constructor in Python?**
   - **"init"** is a special function begin and end with double underscore in Python act as a Constructor.
   - Constructor function will automatically execute when an object of a class is created.

   General format: def __init__(self, [args ……..]): <statements>
5. **What is the purpose of Destructor?**
   - Destructor is also a special method gets executed automatically when an object exit from the scope.
   - In Python, __del__( ) method is used as destructor.

Answer the following questions: (3 Marks)

1. **What are class members? How do you define it?**
   - Variables defined inside a class are called as "Class Variable" and functions are called as "Methods".
   - Class variable and methods are together known as members of the class.
   - The class members should be accessed through objects or instance of class.
   - A class can be defined anywhere in a Python program.

   Syntax for defining a class:

   class class_name:statement_1 statement_2 … statement_n

2. **Write a class with two private class variables and print the sum using a method.**

Coding:

```
class Sum:
   def __init__(self,n1,n2):
      self.__num1=n1
      self.__num2=n2
   def display(self):
      print(self.__num1+self.__num2)
S=Sum(12,14)
S.display()
```

Output: 26

3. **Find the error in the following program to get the given output?**

```
class Fruits:
        def __init__(self, f1, f2):
                self.f1=f1
                self.f2=f2
        def display(self):
                print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
F = Fruits ('Apple', 'Mango')
del F.display
F.display()
```

Output: Fruit 1 = Apple, Fruit 2 = Mango

Error: line 8, in <module> del F.display AttributeError: display

4. **What is the output of the following program?**

```
class Greeting:
def __init__(self, name):
        self.__name = name
def display(self):
        print("Good Morning ", self.__name)
obj=Greeting('Bindu Madhavan')
obj.display()
```

Output: Good Morning Bindu Madhavan

5. **How to define constructor and destructor in Python?**

Constructor:

- **"init"**is a special function begin and end with double underscore in Python act as a Constructor.
- Constructor function will automatically executed when an object of a class is created.

General format of constructor: def __init__(self, [args…..]): <statements>

destructor:

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- In Python, __del__( ) method is used as destructor.

General format of destructor: def __del__(self): <statements>

Answer the following questions: (5 Marks)

1. **Explain about constructor and destructor with suitable example.**

Constructor:

- **"init"**is a special function begin and end with double underscore in Python act as a Constructor.
- Constructor function will automatically executed when an object of a class is created.

General format of constructor: def __init__(self, [args…..]): <statements>

destructor:

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- In Python, __del__( ) method is used as destructor.

General format of destructor: def __del__(self): <statements>

Example:

```
class Sample:
    def __init__(self, num):
        print("Constructor of class Sample...")
        self.num=num
        print("The value is :", num)
    def __del__(self):
        print("Destructor of class Sample…")
S=Sample(10)
```

Output:

Constructor of class Sample...
The value is : 10
Destructor of class Sample…

CHAPTER 11: Database Concepts

Choose the best answer: (1 Mark)

1. What is the acronym of DBMS?
   (A) DataBase Management Symbol (B) Database Managing System
   (C) DataBase Management System (D) DataBasic Management System
2. A table is known as
   (A) tuple (B) attribute **(C) relation** (D) entity
3. Which database model represents parent-child relationship?
   (A) Relational (B) Network **(C) Hierarchical** (D) Object
4. Relational database model was first proposed by
   (A) E F Codd (B) E E Codd (C) E F Cadd (D) E F Codder
5. What type of relationship does hierarchical model represents?
   (A) one-to-one **(B) one-to-many** (C) many-to-one (D) many-to-many
6. Who is called Father of Relational Database from the following?
   (A) Chris Date (B) Hugh Darween
   (C) Edgar Frank Codd (D) Edgar Frank Cadd
7. Which of the following is an RDBMS?
   (A) Dbase (B) Foxpro **(C) Microsoft Access** (D) Microsoft Excel
8. What symbol is used for SELECT statement?
   (A) σ (B) Π (C) X (D) Ω
9. A tuple is also known as
   (A) table **(B) row** (C) attribute (D) field
10. Who developed ER model?
    (A) Chen (B) EF Codd (C) Chend (D) Chand

Answer the following questions: (2 Marks)

1. **Mention few examples of a DBMS.**
   - dbase * Foxpro * MySQL
   - Oracle * FileMakerPro
2. **List some examples of RDBMS.**
   - MySQL * Oracle * SQL Server
   - Microsoft Access * MariaDB * SQLite
3. **What is data consistency?**
   - Data Consistency means that data values are the same at all instances of a database.
4. **What is the difference between Hierarchical and Network data model?**

Hierarchical data model	Network data model
A child record has only one parent node	A child may have many parent nodes.
It represents one-to-one relationship called parent-child relationship in the form of tree structure	It represents the data in many-to-many relationships.

5. **What is normalization?**
   - Normalization is an integral part of RDBMS in order to reduce data redundancy and improve data integrity.

Answer the following questions: (3 Marks)

1. **What is the difference between Select and Project command?**

Select Command	Project Command
The SELECT operation is used for selecting a subset with tuples according to a given condition.	The projection method defines a relation that contains a vertical subset of Relation.
Select filters out all tuples that do not satisfy.	The projection eliminates all attributes of the input relation but those mentioned in the projection list.
Symbol :σ	**Symbol : Π**

2. **What is the role of DBA?**
   - Database Administrator or DBA is the one **who manages the complete database** management system.
   - DBA takes care of the security of the DBMS, managing the license keys, managing user accounts and access etc.
3. **Explain Cartesian Product with a suitable example.**
   - Cross product is a way of combining two relations.
   - The resulting relation contains, both relations being combined.
   - This type of operation is helpful to merge columns from two relations.

   Example: A x B means A times B, where the relation A and B have different attributes.

4. **Explain Object Model with example.**
   - Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
   - This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.

   An **example** of the Object model is **Shape, Circle, Rectangle** and **Triangle** are all objects in this model.
      - **Circle** has the attribute **radius.**
      - **Rectangle** has the attributes **length and breadth.**
      - **Triangle** has the attributes **base and height**.
      - The objects Circle, Rectangle and Triangle **inherit** from the object Shape.
5. **Write a note on different types of DBMS users.**
   - **Database Administrators:** Database Administrator or DBA is the one who manages the complete database management system.
   - **Application Programmers or Software Developers:** This user group is involved in developing and designing the parts of DBMS.
   - **End User:** End users are the one who store, retrieve, update and delete data.
   - **Database designers:** They are responsible for identifying the data to be stored in the database for choosing appropriate structures to represent and store the data.

Answer the following questions: (5 Marks)

1. Explain the different types of data model.

i) Hierarchical Model:

- In Hierarchical model, data is represented as a simple tree like structure form. This model represents a one-to-many relationship ie parent-child relationship. This model is mainly used in IBM Main Frame computers.

ii) Relational Model:

- The Relational Database model was first proposed by E.F. Codd in 1970.
- The basic structure of data in relational model is tables (relations).
- All the information's related to a particular type is stored in rows of that table.

iii) Network Model:

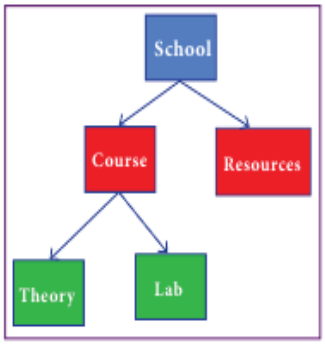
- In a Network model, a child may have many parent nodes.
- It represents the data in many-to-many relationships.
- This model is easier and faster to access the data.

iv) Entity Relationship Model. (ER model):

- In this database model, relationships are created by dividing the object into entity and its characteristics into attributes. It was developed by Chen in 1976.

v) Object Model:

- Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
- This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.



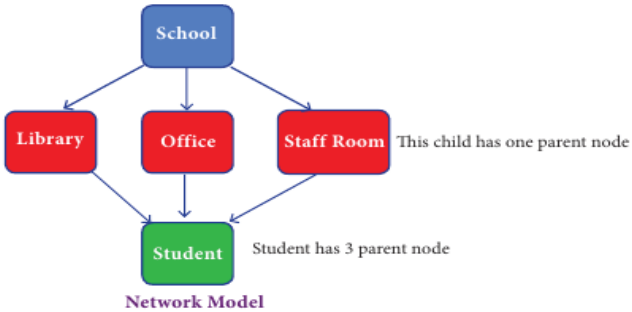
Hierarchical Model Fig. 11.3

Stu_id	Name	Age
1	Malar	17
2	Suncar	16
3	Velu	16

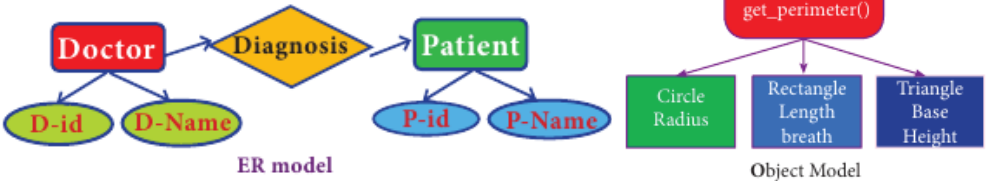
Subj_id	Name	Teacher
1	C++	Kannan
2	Php	Ramakrishnan
3	Python	Vidhya

Stu_id	Subj_id	Marks
1	1	92
1	2	89
3	2	96

Relational Model



Network Model



ER model

Object Model

2. Explain the different types of relationship mapping.

Types of Relationships:

1. One-to-One Relationship
2. One-to-Many Relationship
3. Many-to-One Relationship
4. Many-to-Many Relationship

- **One-to-One Relationship:** In One-to-One Relationship, one entity is related with only one other entity

For Example: A student can have only one exam number.

- **One-to-Many Relationship:** In One-to-Many relationship, one entity is related to many other entities.

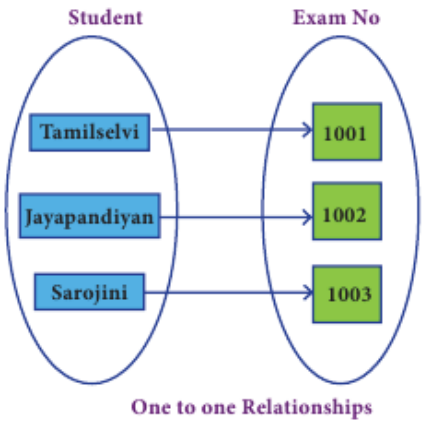
For Example: One Department has many staff members.

- **Many-to-One Relationship:** In Many-to-One Relationship, many entities can be related with only one in the other entity.

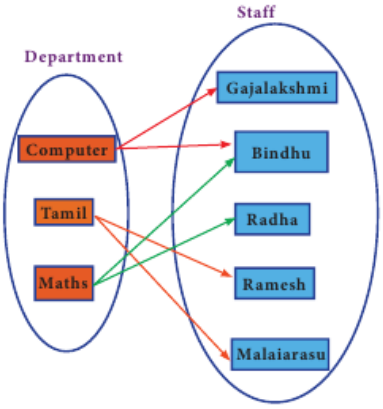
For Example: A number of staff members working in one Department

- **Many-to-Many Relationship:** A many-to-many relationship occurs when multiple records in a table are associated with multiple records in another table.

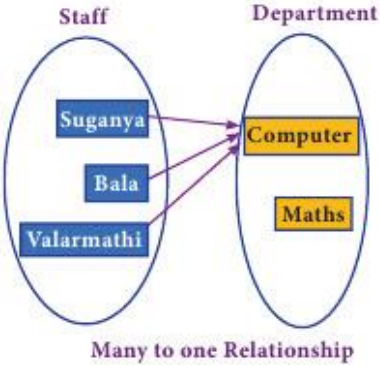
For Example: Many Books in a Library are issued to many students.



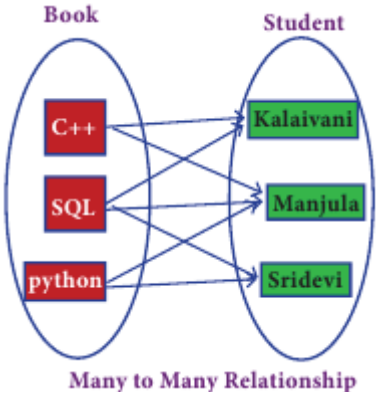
One to one Relationships



One to Many Mapping



Many to one Relationship



Many to Many Relationship

3. Differentiate DBMS and RDBMS.

Basis of Comparison	DBMS	RDBMS
Expansion	Database Management System	Relational DataBase Management System
Data storage	Navigational model (ie) data by linked records	Relational model (ie) data in tables as row and column
Data redundancy	Present	Not Present
Normalization	Not performed	RDBMS uses normalization to reduce redundancy
Data access	Consumes more time	Faster, compared to DBMS.
Keys and indexes	Does not use.	used to establish relationship. Keys are used in RDBMS.
Transaction management	Inefficient, Error prone and insecure.	Efficient and secure.
Distributed Databases	Not supported	Supported by RDBMS.
Example	Dbase, FoxPro.	SQL server, Oracle, mysql, MariaDB, SQLite, MS Access.

4. Explain the characteristics of RDBMS.

Ability to manipulate data	RDBMS provides the facility to manipulate data (store, modify and delete)in a data base.
Reduced Redundancy	RDBMS follows Normalisation which divides the data in such a way that repetition is minimum.
Data Consistency	On live data, it is being continuously updated and added, maintaining the consistency of data can become a challenge. But RDBMS handles it by itself.
Support Multiple user and Concurrent Access	RDBMS allows multiple users to work on it, at the same time and still manages to maintain the data consistency.
Query Language	RDBMS provides users with a simple query language, using which data can be easily fetched, inserted, deleted and updated in a database.
Security	we can easily secure our data by restricting user access.
DBMS Supports Transactions	It allows us to better handle and manage data integrity in real world applications.

5. Explain the different operators in Relational algebra with suitable examples.

Relational Algebra:

(i) Union(Symbol: ∪) (ii) Intersection(Symbol: ∩)

(iii) Difference(Symbol: -) (iv) Cartesian Product(Symbol: X)

UNION (Symbol :∪) AUB

- It includes all tuples that are in tables A or in B. It also eliminates duplicates.

INTERSECTION (symbol : ∩) A ∩ B

- Defines a relation consisting of a set of all tuple that are in both in A and B. However, A and B must be union-compatible.

DIFFERENCE (Symbol : -)

- The result of A – B, is a relation which includes all tuples that are in A but not in B. The attribute name of A has to match with the attribute name in B.

CARTESIAN PRODUCT (Symbol : X)

- Cross product is a way of combining two relations. The resulting relation contains, both relations being combined.

Example:

Table A	
Stuno	Name
Cs1	Kannan
Cs3	Lenin
Cs4	Raja

Table B	
Stuno	Name
Cs1	Kannan
Cs2	Shankar
Cs3	Lenin

Table A ∪ B	
Stuno	Name
Cs1	Kannan
Cs2	Shankar
Cs3	Lenin
Cs4	Raja

Table A ∩ B	
Stuno	Name
Cs1	Kannan
Cs3	Lenin

Table A – B	
Stuno	Name
Cs4	Raja

Table A	
Stuno	Name
Cs1	Kannan
Cs2	Shankar

Table B	
Course	Subject
Cs28	Big Data
Cs62	R Language

Table A × B			
Stuno	Name	Course	Subject
Cs1	Kannan	Cs28	Big Data
Cs1	Kannan	Cs62	R Language
Cs2	Shankar	Cs28	Big Data
Cs2	Shankar	Cs62	R Language

Choose the best answer: (1 Mark)

1. Which commands provide definitions for creating table structure, deleting relations, and modifyingrelation schemas.
   (A) DDL (B) DML (C) DCL (D) DQL
2. Which command lets to change the structure of the table?
   (A) SELECT (B) ORDER BY (C) MODIFY **(D) ALTER**
3. The command to delete a table including the structure is
   (A) DROP (B) DELETE (C) DELETE ALL D) ALTER TABLE
4. Queries can be generated using
   (A) SELECT (B) ORDER BY (C) MODIFY (D) ALTER
5. The clause used to sort data in a database
   (A) SORT BY **(B) ORDER BY** (C) GROUP BY (D) SELECT

Answer the following questions: (2 Marks)

1. **Write a query that selects all students whose age is less than 18 in order wise.**
   SELECT * FROM Student WHERE Age<=18 ORDER BY Name;
2. **Differentiate Unique and Primary Key constraint.**

Unique Key Constraint	Primary Key Constraint
This constraint ensures that no two rows have the same value in the specified columns.	Primary constraint declares a field as a Primary key which helps to uniquely identify a record.

3. Write the difference between table constraint and column constraint?

Table Constraint	Column Constraint
Table constraints apply to a group of one or more columns	Column constraints apply only to individual column.

4. Which component of SQL lets insert values in tables and which lets to create a table?
   - DML: **Insert** - Insert data into a table
   - DDL: **Create** - To create tables in the database.
5. What is the difference between SQL and MySQL?

SQL	MySQL
Structured Query Language is a language used for accessing databases.	**MySQL** is a Relational database management system.

Answer the following questions: (3 Marks)

1. **What is a constraint? Write short note on Primary key constraint.**
   - Constraint is a condition applicable on a field or set of fields.
   - Primary constraint declares a field as a Primary key which helps to uniquely identify a record.
   - The primary key does not allow **NULL** values and therefore a primary key field must have the **NOT NULL** constraint.
2. **Write a SQL statement to modify the student table structure by adding a new field.**
   - To add a new column "Address" of type "char" to the Student table, the command is**:**
     ALTER TABLE Student ADD Address char;
3. **Write any three DDL commands.**
   - **Create:** To create tables in the database.
   - **Alter:** Alters the structure of the database.
   - **Drop:** Delete tables from database.
4. **Write the use of Savepoint command with an example.**
   - The **SAVEPOINT** command is used to temporarily save a transaction so that you can rollback to the point whenever required.

   Syntax: SAVEPOINT savepoint_name;
   Example: UPDATE student SET Name = 'Mini' WHERE Admno=105;
   SAVEPOINT A;

5. **Write a SQL statement using DISTINCT keyword.**
   - The **DISTINCT** keyword is used along with the **SELECT** command to eliminate duplicate rows in the table.

   For Example: SELECT DISTINCT Place FROM Student;

Answer the following questions: (5 Marks)

1. **Write the different types of constraints and their functions.**

   Constraint is a condition applicable on a field or set of fields.

   Type of Constraints:
   - **Unique Constraint:** This constraint ensures that no two rows have the same value in the specified columns.
   - **Primary Key Constraint:** This constraint declares a field as a Primary key which helps to uniquely identify a record. The primary key does not allow **NULL** values and therefore a field declared as primary key must have the **NOT NULL** constraint.
   - **Default constraint**: This constraint is used to assign a default value for the field. When no value is given for the specified field having **DEFAULT** constraint, automatically the default value will be assigned to the field.
   - **Check Constraint:** This constraint helps to set a limit value placed for a field. When we define a check constraint on a single column, it allows only the restricted values on that field.
   - **Table Constraint:** When the constraint is applied to a group of fields of the table, it is known as Table constraint. The table constraint is normally given at the end of the table definition.

2. **Consider the following employee table. Write SQL commands for the qtns.(i) to (v).**

EMP CODE	NAME	DESIG	PAY	ALLO WANCE
S1001	Hariharan	Supervisor	29000	12000
P1002	Shaji	Operator	10000	5500
P1003	Prasad	Operator	12000	6500
C1004	Manjima	Clerk	8000	4500
M1005	Ratheesh	Mechanic	20000	7000

(i) To display the details of all employees in descending order of pay.

SELECT * FROM employee ORDER BY DESC;

(ii) To display all employees whose allowance is between 5000 and 7000.

SELECT * FROM employee WHERE allowance BETWEEN 5000 AND 7000;

(iii) To remove the employees who are mechanic.

DELETE FROM employee WHERE desig=”Mechanic”;

(iv) To add a new row.

INSERT INTO employee (empcode, name, desig, pay, allowance) VALUES(S1002,Baskaran,Supervisor,29000,12000);

(v) To display the details of all employees who are operators.

SELECT * FROM employee WHERE design=“Operator“;

3. **What are the components of SQL? Write the commands in each Components of SQL:**

DATA DEFINITION LANGUAGE (DDL):

- **Create:** To create tables in the database.
- **Alter:** Alters the structure of the database.
- **Drop:** Delete tables from database.
- **Truncate:** Remove all records from a table, also release the space occupied by those records.

DATA MANIPULATION LANGUAGE (DML) :

- **Insert**: Inserts data into a table
- **Update:** Updates the existing data within a table.
- **Delete**: Deletes all records from a table, but not the space occupied by them.

DATA CONTROL LANGUAGE (DCL):

- **Grant:** Grants permission to one or more users to perform specific tasks.
- **Revoke:** Withdraws the access permission given by the GRANT statement.

TRANSACTIONAL CONTROL LANGUAGE (TCL):

- **Commit:** Saves any transaction into the database permanently.
- **Roll back:** Restores the database to last commit state.
- **Save point:** Temporarily save a transaction so that you can rollback.

DATA QUERY LANGUAGE (DQL):

- **Select:** It displays the records from the table.

4. **Construct the following SQL statements in the student table-**

(i) SELECT statement using GROUP BY clause.

(ii) SELECT statement using ORDER BY clause.

(i) SELECT statement using GROUP BY clause.

The **GROUP BY** clause is used with the **SELECT** statement to group the students on rows or columns having identical values or divide the table in to groups.

For example: The number of male students or female students of a class, the **GROUP BY** clause may be used. The command is,

SELECT Gender FROM Student GROUP BY Gender;

(ii) SELECT statement using ORDER BY clause.

The **ORDER BY** clause in SQL is used to sort the data in either ascending or descending based on one or more columns.

- By default **ORDER BY** sorts the data in ascending order.
- We can use the keyword **DESC** to sort the data in descending order and the keyword **ASC** to sort in ascending order.

For example: To display the students in alphabetical order of their names, the command is used as,

SELECT * FROM student WHERE Age>=18 ORDER BY Name DESC;

5. **Write a SQL statement to create a table for employee having any five fields and create a table constraint for the employee table.**

SQL Statement:

```
CREATE TABLE employee
        (empno integer NOT NULL,
        name char(20),
        desig char(20),
        pay integer,
        allowance integer,
        PRIMARY KEY (empno));
```

CHAPTER 13: Python and CSV files

Choose the best answer: (1 Mark)

1. A CSV file is also known as a ….
   (A) Flat File (B) 3D File (C) String File (D) Random File
2. The expansion of CRLF is
   (A) Control Return and Line Feed (B) Carriage Return and Form Feed
   (C) Control Router and Line Feed **(D) Carriage Return and Line Feed**
3. Which of the following module is provided by Python to do several operations on the CSV files?
   (A) py (B) xls **(C) csv** (D) os
4. Which of the following mode is used when dealing with non-text files like image or exe files?
   (A) Text mode **(B) Binary mode** (C) xls mode (D) csv mode 5
5. The command used to skip a row in a CSV file is
   (A) next() (B) skip() (C) omit() (D) bounce()
6. Which of the following is a string used to terminate lines produced by writer()method of csvmodule?
   (A) Line Terminator (B) Enter key (C) Form feed (D) DataTerminator
7. What is the output of the following program?

```
import csv
d=csv.reader(open('c:\PYPRG\ch13\city.csv'))
next(d)
for row in d:
print(row)
```

   if the file called "city.csv" contain the following details

```
chennai, mylapore
 mumbai, andheri
```

   (A) chennai,mylapore **(B) mumbai,andheri**
   (C) Chennai (D) chennai,mylapore
   Mumbai mumbai,andheri
8. Which of the following creates an object which maps data to a dictionary?
   (A) listreader() (B) reader() (C) tuplereader() **(D) DictReader ()**
9. Making some changes in the data of the existing file or adding more data is called
   (A) Editing (B) Appending **(C) Modification** (D) Alteration
10. What will be written inside the file test.csv using the following program.

```
Import csv
D=[['Exam'],['Quarterly'],['Halfyearly']]
Csv.register_dialect('M',lineterminator='\n')
        Wr=csv.writer(f,dialect='M')
        Wr.writerows(D)
f.close( )
```

    (A) Exam Quarterly Halfyearly (B) Exam Quarterly Halfyearly
    (C) E **(D) Exam,**
    Q **Quarterly,**
    H **Halfyearly**

Answer the following questions: (2 Marks)

1. **What is CSV File?**
   - A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.
2. **Mention the two ways to read a CSV file using Python.**
   The two ways to read a CSV file are:
   - using csv.reader() function * using DictReader class.
3. **Mention the default modes of the File.**
   - The default is reading ('r') in text mode.
   - In this mode, while reading from the file the data would be in the format of **strings.**
4. **What is use of next() function?**
   - **'next()' command** is used to avoid or skip the first row or row heading.

   Example: While sorting the row heading is also get sorted, to avoid that the first is skipped using next().
5. **How will you sort more than one column from a csv file?Give an example statement.**
   - To sort by more than one column you can use **itemgetter** with multiple indices.

   Syntax:
   sortedlist = sorted(data, key=operator.itemgetter(Col_number),reverse=True)
   Example: sortedlist = sorted (data, key=operator.itemgetter(1))

Answer the following questions: (3 Marks)

1. **Write a note on open() function of python. What is the difference between the two methods?**
   - Python has a built-in function **open()** to open a file.
   - This function is used to read or modify the file accordingly.

   Method 1: f = open("test.txt")
   f.close()
   Method 2: with open("test.txt",'r') as f:
   - The first method is **not entirely safe.** If an **exception** occurs when you are performing some operation with the file, the code exits without closing the file.
   - The best way to do this is using the **"with"** statement. This ensures that the file is closed when the block inside **with** is exited
2. **Write a Python program to modify an existing file.**
   Coding:

```
import csv
row = ['3', 'Meena','Bangalore']
with open('student.csv', 'r') as readFile:
        reader = csv.reader(readFile)
        lines = list(reader)
        lines[3] = row
with open('student.csv', 'w') as writeFile:
    writer = csv.writer(writeFile)
    writer.writerows(lines)
readFile.close()
writeFile.close()
```

3. **Write a Python program to read a CSV file with default delimiter comma (,).**

Coding:

```
import csv
with open('c:\pyprg\sample1.csv','r') as F:
    reader = csv.reader(F)
    for row in reader:
        print(row)
F.close()
```

4. **What is the difference between the write mode and append mode.**

w - Write Mode	a - Append Mode
Open a file for writing.	Open for appending at the end of the file without truncating it.
Creates a new file if it does not exist or truncates the file if it exists.	Creates a new file if it does not exist.

5. **What is the difference between reader()method and DictReader() class?**

Reader():	DictReader():
The reader function is designed to take each line of the file and make a list of all columns.	DictReader function creates an object which maps data to a dictionary. It works by reading the first line of the CSV and using each comma separated value in this line as a dictionary key.
csv. Reader work with list/tuple.	csv.DictReader work with dictionary.

Answer the following questions: (5 Marks)

1. **Differentiate Excel file and CSV file.**

Excel	CSV
Excel is a binary file that holds information about all the worksheets in a file, including both content and formatting	CSV format is a plain text format with a series of values separated by commas.
XLS files can only be read by applications that have been especially written to read their format.	CSV can be opened with any text editor in Windows like notepad, MS Excel, OpenOffice, etc.
Excel file saves the file in .xls or .xlsx format.	Saves the file with the extension .csv
Excel consumes more memory while importing data	Importing CSV files can be much faster, and it also consumes less memory

2. **Tabulate the different mode with its meaning.**

Mode	Description
'r'	Open a file for reading. (default)
'w'	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)

3. **Write the different methods to read a File in Python.**

csv.reader() function:

- Contents of CSV file can be read with the help of **csv.reader()** method.
- The reader function is designed to take each line of the file and make a list of all columns.

Syntax: csv.reader(fileobject,delimiter,fmtparams)

where

- file object : passes the path and the mode of the file
- delimiter : an optional parameter containing the standard dialects.
- fmtparams: optional parameter which help to override the default values of the dialects.

Example:

```
import csv
with open('c:\\pyprg\\sample1.csv', 'r') as F:
        reader = csv.reader(F)
        for row in reader:
                print(row)
        F.close()
```

Output:

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

DictReader class:

- It creates an object which maps data to a dictionary.
- **DictReader** works by reading the first line of the CSV and using each comma separated value in this line as a **dictionary key.**

Example:

```
import csv
filename = 'c:\\pyprg\\sample8.csv'
input_file =csv.DictReader(open(filename,'r'))
for row in input_file:
        print(dict(row))
```

Output:

```
{'ItemName ': 'Keyboard ', 'Quantity': '48'}
{'ItemName ': 'Monitor', 'Quantity': '52'}
{'ItemName ': 'Mouse ', 'Quantity': '20'}
```

4. Write a Python program to write a CSV File with custom quotes.

Coding:

```
import csv
csv.register_dialect('myDialect', delimiter = '|', quoting=csv.QUOTE_ALL)
with open('c:\pyprg\grade.csv', 'w') as csvfile:
    fieldnames = ['Name', 'Grade']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames,
        dialect="myDialect")
           writer.writeheader()
        writer.writerows([{'Grade': 'B', 'Name': 'Anu'},
                          {'Grade': 'A', 'Name': 'Beena'},
                          {'Grade': 'C', 'Name': 'Tarun'}])
print("writing completed")
```

5. Write the rules to be followed to format the data in a CSV file.

Rules to be followed to format data in a CSV file

- Each row of data is to be located on a separate line, delimited by a line break by pressing enter key.
- The last record in the file may or may not have an ending line break.
- There may be an optional header line appearing as the first line of the file
- Within the header and each record, there may be one or more fields, separated by commas.
- Each field may or may not be enclosed in double quotes.
- Fields containing line breaks, double quotes, and commas should be enclosed in double-quotes.
- If double-quotes are used to enclose fields, then a double-quote appearing inside a field must be preceded with another double quote.

CHAPTER 14: Importing C++ programs in Python

Choose the best answer: (1 Mark)

1. Which of the following is not a scripting language?
   (A) JavaScript (B) PHP (C) Perl **(D) HTML**
2. Importing C++ program in a Python program is called
   (A) wrapping (B) Downloading (C) Interconnecting (D) Parsing
3. The expansion of API is
   (A) Application Programming Interpreter
   (B) Application Programming Interface
   (C) Application Performing Interface
   (D) Application Programming Interlink
4. A framework for interfacing Python and C++ is
   (A) Ctypes (B) SWIG (C) Cython **(D) Boost**
5. Which of the following is a software design technique to split your code into separate parts?
   (A) Object oriented Programming **(B) Modular programming**
   (C) Low Level Programming (D) Procedure oriented Programming
6. The module which allows you to interface with the Windows operating system is
   (A) OS module (B) sys module (C) csv module (D) getopt module
7. getopt() will return an empty array if there is no error in splitting strings to
   (A) argv variable (B) opt variable **(C) args variable** (D) ifile variable
8. Identify the function call statement in the following snippet.

```
if __name__ =='__main__':
main(sys.argv[1:])
```

   (A) main(sys.argv[1:]) **(B) __name__** (C) __main__ (D) argv
9. Which of the following can be used for processing text, numbers, images, and scientific data?
   (A) HTML (B) C (C) C++ **(D) PYTHON**
10. What does __name__ contains ?
    (A) c++ filename (B) main() name
    (C) python filename (D) os module name

Answer the following questions: (2 Marks)

1. **What is the theoretical difference between Scripting language and other programming language?**

Scripting language	Programming language
Scripting languages do not require the compilation step and are rather interpreted.	Programming language needs to be compiled before running.
A scripting language requires an interpreter. Ex: pyhon	A Programming language requires a compiler. Ex: C++

2. **Differentiate compiler and interpreter.**

Compiler	Interpreter
Compiler reads entire program for compilation.	Interpreter reads single statement at a time for interpretation.
Error deduction is difficult.	Error deduction is easy.
Example: c++	**Example:** Python

3. **Write the expansion of (i) SWIG (ii) MinGW**
   - **SWIG** - Simplified Wrapper interface Generator
   - **MinGW** - Minimalist GNU for Windows.
4. **What is the use of modules?**
   - Modules are used to break down large programs into small manageable and organized files.
   - Modules provide reusability of code.
   - We can define our most used functions in a module and import it, instead of copying their definitions into different programs.
5. **What is the use of cd command. Give an example.**
   - **'cd'** command used to change directory and absolute path refers to the complete path where Python is installed.

   Syntax: cd <absolute path>

   Example: c:\>cd c:\ program files \ openoffice 4 \ program

Answer the following questions: (3 Marks)

1. **Differentiate PYTHON and C++**

PYTHON	C++
Python is an interpreter based language.	C++ is a compiler based language.
Python is interpreted dynamically	C++ is compiled statically
Data type is not required while declaring variable	Data type is required while declaring variable
It can act both as scripting and general purpose language	It is a general purpose language

2. **What are the applications of scripting language?**
   - To automate certain tasks in a program.
   - Extracting information from a data set.
   - Less code intensive as compared to traditional programming language.
   - Can bring new functions to applications and glue complex systems together.
3. **What is MinGW? What is its use?**
   - MinGw-W64 (version of MinGW) is the best compiler for C++ on Windows.
   - MinGW allows to compile and execute C++ program dynamically through Python program using g++.
   - Python program that contains the C++ coding can be executed through either by using command prompt or by using run terminal.
4. **Identify the module ,operator, definition name for the following.**

   welcome.display()

   Welcome - Module name
   . - Dot operator
   display() - Function call
5. **What is sys.argv? What does it contain?**
   - sys.argv is the list of command-line arguments passed to the Python program.
   - It's basically a list holding the command-line arguments of the program.
   - To use **sys.argv**, **import sys** should be used.
   - The first argument, sys.argv[0] contains the name of the python program (example pali.py)
   - sys.argv [1]is the next argument passed to the program (here it is the C++ file), which will be the argument passed through main ().

Answer the following questions: (5 Marks)

1. **Write any 5 features of Python.**
   - Python uses Automatic Garbage Collection whereas C++ does not.
   - C++ is a statically typed language, while Python is a dynamically typed language.
   - Python runs through an interpreter, while C++ is pre-compiled.
   - Python code tends to be 5 to 10 times shorter than that written in C++.
   - In Python, there is no need to declare types explicitly where as it should be done in C++
   - In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand. Whereas in C++ return statement can return only one value.

2. **Explain each word of the following command.**
   Python <filename.py> -<i> <C++ filename without cpp extension>

Python	keyword to execute the Python program from command-line
filename.py	Name of the Python program to executed
- i	input mode
C++ filename without cpp extension	name of C++ file to be compiled and executed

3. **What is the purpose of sys,os,getopt module in Python.Explain**

Python's sys module:

- sys module provides access to some variables used by the interpreter and to functions that interact with the interpreter.

Python's OS Module:

- The ***OS*** module in Python provides a way of using operating system dependent functionality.
- The functions that the ***OS*** module allows you to interface with the Windows operating system where Python is running on.

Python getopt module:

- The getopt module of Python helps you to parse (split) command-line options and arguments.
- This module provides getopt() method to enable command-line argument parsing.

4. **Write the syntax for getopt() and explain its arguments and return values**

Syntax: **<opts>,<args>=getopt.getopt(argv, options, [long_options])**

The detail of the parameters:

- **argv** − This is the argument list of values to be splited. In our program the complete command will be passed as a list.
- **options** − This is string of option letters that the Python program recognize as, for input or for output.
- **long_options** −This contains a list of strings.

returns value:

- getopt() method returns value consisting of two elements.
- Each of these values are stored separately in two different list (arrays) **opts** and **args** .
- Opts contains list of splitted strings like mode and path.
- **args** contains error string, if at all the comment is given with wrong path or mode.

5. **Write a Python program to execute the following c++ coding**

#include <iostream>
using namespace std;
int main()
{ cout<<“WELCOME”;
return(0);
}

The above C++ program is saved in a file welcome.cpp

c++ coding:

```
import sys, os, getopt
def main(argv):
        opts, args = getopt.getopt(argv, "i:")
        for o, a in opts:
                if o in "-i":
                        run(a)
def run(a):
        inp_file=a+'.cpp'
        exe_file=a+'.exe'
        os.system('g++ ' + inp_file + ' -o ' + exe_file)
        os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

CHAPTER 15: Data manipulation through SQL

Choose the best answer: (1 Mark)

1. Which of the following is an organized collection of data?
   (A) Database (B) DBMS (C) Information (D) Records
2. SQLite falls under which database system?
   (A) Flat file database system **(B) Relational Database system**
   (C) Hierarchical database system (D) Object oriented Database system
3. Which of the following is a control structure used to traverse and fetch the records of the database?
   (A) Pointer (B) Key **(C) Cursor** (D) Insertion point
4. Any changes made in the values of the record should be saved by the command
   (A) Save (B) Save As **(C) Commit** (D) Oblige
5. Which of the following executes the SQL command to perform some action?
   (A) execute() (B) key() (C) cursor() (D) run()
6. Which of the following function retrieves the average of a selected column of rows in a table?
   (A) Add() (B) SUM() **(C) AVG()** (D) AVERAGE()
7. The function that returns the largest value of the selected column is
   (A) MAX() (B) LARGE() (C) HIGH() (D) MAXIMUM()
8. Which of the following is called the master table?
   (A) sqlite_master (B) sql_master (C) main_master (D) master_main
9. The most commonly used statement in SQL is
   (A) cursor **(B) select** (C) execute (D) commit
10. Which of the following clause avoid the duplicate?
    (A) Distinct (B) Remove (C) Where (D) GroupBy

Answer the following questions: (2 Marks)

1. **Mention the users who uses the Database.**
   - Users of database can be human users, other programs or applications
2. **Which method is used to connect a database? Give an example.**
   - Create a connection using **connect () method** and pass the name of the database File.

   Example: connection = sqlite3.connect ("Academy.db")
3. **What is the advantage of declaring a column as "INTEGER PRIMARY KEY"**
   - If a column of a table is declared to be an **INTEGER PRIMARY KEY**, then whenever a NULL will be used as an input for this column, the **NULL will be automatically converted into an integer** which will one larger than the highest value so far used in that column.
4. **Write the command to populate record in a table. Give an example.**
   - To populate (add record) the table "INSERT" command is passed to SQLite.

     Example: INSERT INTO Student (Rollno, Name)
     VALUES (101, "Akshay");
5. **Which method is used to fetch all rows from the database table?**
   - The **fetchall()** method is used to fetch all rows from the database table.

   Example: result = cursor.fetchall()

Answer the following questions: (3 Marks)

1. **What is SQLite?What is it advantage?**
   - SQLite is a simple relational database system.

   Advantage:
   - SQLite is fast, rigorously tested, and flexible, making it easier to work.
   - Python has a native library for SQLite.
2. **Mention the difference between fetchone() and fetchmany()**

fetchone()	fetchmany()
The **fetchone()** method returns the next row of a query result set or None in case there is no row left.	This method returns the number of rows of the result set.
Using while loop and fetchone() method we can display all the records from a table.	Displaying specified number of records is done by using fetchmany(n).

3. **What is the use of Where Clause.Give a python statement Using the where clause.**
   - The WHERE clause is used to extract only those records that fulfill a specified condition.

   EXAMPLE: To display the different grades scored by male students from "student table"

   cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")

4. **Read the following details.Based on that write a python script to display department wise records**

database name :- organization.db

Table name :- Employee

Columns in the table :- Eno, EmpName, Esal, Dept

Coding:

```
import sqlite3
connection=sqlite3.connect("organization.db")
c = connection.execute("SELECT * FROM Employee GROUP BY  dept")
for row in c:
    print(row)
connection.close( )
```

5. **Read the following details.Based on that write a python script to display records in desending order of Eno**

database name :- organization.db

Table name :- Employee

Columns in the table :- Eno, EmpName, Esal, Dept

Coding:

```
import sqlite3
connection = sqlite3.connect("organization.db")
cursor=connection.cursor()
cursor.execute("SELECT * FROM Employee ORDER BY Eno DESC")
result=cursor.fetchall()
print(result)
```

Answer the following questions: (5 Marks)

1. **Write in brief about SQLite and the steps used to use it.**

- SQLite is a simple relational database system, which saves its data in regular data files within internal memory of the computer.
- It is designed to be embedded in applications, instead of using a separate database server program such as MySQL or Oracle.

Advantages:

- SQLite is fast, rigorously tested, and flexible, making it easier to work.
- Python has a native library for SQLite. To use SQLite,

To use SQLite,

Step1: import sqlite3

Step2: Create a connection using connect () method and pass the name of the database File

Step3: Set the cursor object cursor = connection. cursor ()

Example:

```
Import sqlite3
Connection = sqlite3.connect ("Academy.db")
Cursor = connection.cursor( )
```

2. **Write the Python script to display all the records of the following table using fetchmany()**

Icode	ItemName	Rate
1003	Scanner	10500
1004	Speaker	3000
1005	Printer	8000
1008	Monitor	15000
1010	Mouse	700

Coding:

```
import sqlite3
connection=sqlite3.connect("company.db")
cursor=connection.cursor( )
cursor.execute("""DROP TABLE Product;""")
sql_command="""CREATE TABLE Product(Icode INTEGER PRIMARY KEY,
Item_Name VARCHAR(20), Rate INTEGER);"""
cursor.execute(sql_command)
sql_command="""INSERT INTO Product VALUES (1003,"Scanner",10500);"""
cursor.execute(sql_command)
sql_command="""INSERT INTO Product VALUES(1004,"Speaker",3000);"""
cursor.execute(sql_command)
sql_command="""INSERT INTO Product VALUES(1005,"Printer",8000);"""
cursor.execute(sql_command)
sql_command="""INSERT INTO Product VALUES(1008,"Moniter",15000);"""
cursor.execute(sql_command)
sql_command="""INSERT INTO Product VALUES(1010,"Mouse",700);"""
cursor.execute(sql_command)
connection.commit( )
cursor.execute("SELECT * FROM Product")
ans=cursor.fetchmany(5)
for i in ans:
    print(i)
connection.close( )
```

Output:

```
(1003, 'Scanner', 10500)
(1004, 'Speaker', 3000)
(1005, 'Printer', 8000)
(1008, 'Moniter', 15000)
(1010, 'Mouse', 700)
```

3. What is the use of HAVING clause. Give an example python script

- Having clause is used to filter data based on the group functions.
- This is similar to WHERE condition but can be used only with group functions.
- Group functions cannot be used in WHERE Clause but can be used in HAVING clause.

Example:

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student GROUP BY GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

Output: ['gender','COUNT(GENDER)']
[('M', 5)]

4. Write a Python script to create a table called ITEM with following specification.

Add one record to the table.
Name of the database :- ABC
Name of the table :- Item
Column name and specification :-

Icode	**:-**	**integer and act as primary key**
Item Name	**:-**	**Character with length 25**
Rate	**:-**	**Integer**
Record to be added	**:-**	**1008, Monitor,15000**

Coding:

```
import sqlite3
connection=sqlite3.connect ("organization.db")
cursor=connection.cursor( )
cursor.execute ("""DROP TABLE item;""")
sql_command="""CREATE TABLE item(Icode INTEGER PRIMARY KEY, Item_Name
VARCHAR(25),Rate INTEGER);"""
cursor.execute (sql_command)
sql_command="""INSERT INTO item VALUES(1008,"Monitor",15000);"""
cursor.execute (sql_command)
connection.commit( )
cursor.execute ("SELECT * FROM item")
ans=cursor.fetchall ( )
for i in ans:
    print(i)
connection.close ( )
```

Output: **(1008, 'Monitor', 15000)**

5. **Consider the following table Supplier and item .Write a python script for (i) to (ii)**

SUPPLIER				
Suppno	**Name**	**City**	**Icode**	**SuppQty**
S001	**Prasad**	**Delhi**	**1008**	**100**
S002	**Anu**	**Bangalore**	**1010**	**200**
S003	**Shahid**	**Bangalore**	**1008**	**175**
S004	**Akila**	**Hydrabad**	**1005**	**195**
S005	**Girish**	**Hydrabad**	**1003**	**25**
S006	**Shylaja**	**Chennai**	**1008**	**180**
S007	**Lavanya**	**Mumbai**	**1005**	**325**

Coding:

```
import sqlite3
connection=sqlite3.connect("company.db")
cursor=connection.cursor( )
cursor.execute ("""DROP TABLE Supplier;""")
sql_command="""CREATE TABLE Supplier(Suppno INTEGER, Name VARCHAR(25),
City VARCHAR(20), Icode INTEGER, SuppQty INTEGER);"""
cursor.execute(sql_command)
Supplier_data =[("S001","Prasad","Delhi","1008","100"),
        ("S002","Anu","Bangalore","1010","200"),
        ("S003","Shahid","Bangalore","1008","175"),
        ("S004","Ahila","Hydrabad","1005","195"),
        ("S005","Girish","Hydrabad","1003","25"),
        ("S006","Shylaja","Chennai","1008","180")
("S007","Lavanya","Mumbai","1005","325")]
for p in Supplier_data:
   format_str = """INSERT INTO Supplier (Suppno, Name, City, Icode, SuppQty)
   VALUES ("{sno}","{name}", "{city}", "{code}", "{qty}");"""
   sql_command=(format_str.format(sno=p[0],name=p[1],city=p[2],code=p[3],qty=p[4]))
   cursor.execute(sql_command)
connection.commit( )
cursor.execute("SELECT * FROM Supplier WHERE NOT City ='Delhi'")
print("Display All Suppliers not in Delhi")
ans=cursor.fetchall( )
for i in ans:
   print(i)
cursor.execute("UPDATE Supplier SET SuppQty = SuppQty + 40 WHERE Name
='Ahila'")
print("Update Ahila SuppQty")
cursor.execute("SELECT * FROM Supplier")
ans=cursor.fetchall( )
for i in ans:
   print(i)
connection.close ( )
```

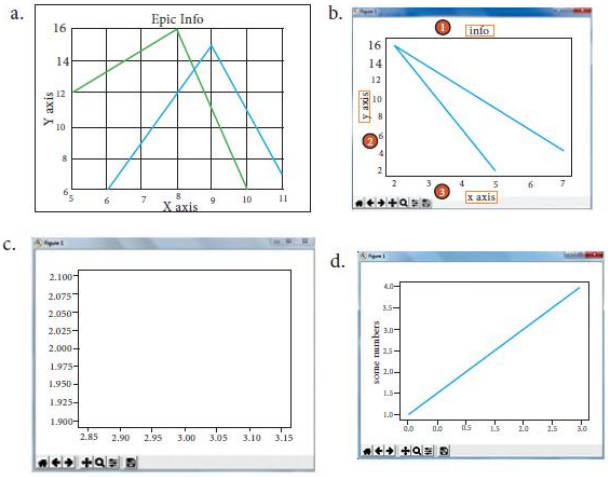
Output:

```
Display All Suppliers not in Delhi
('S002', 'Anu', 'Bangalore', 1010, 200)
('S003', 'Shahid', 'Bangalore', 1008, 175)
('S004', 'Ahila', 'Hydrabad', 1005, 195)
('S005', 'Girish', 'Hydrabad', 1003, 25)
('S006', 'Shylaja', 'Chennai', 1008, 180)
('S007', 'Lavanya', 'Mumbai', 1005, 325)
Update Ahila SuppQty
('S001', 'Prasad', 'Delhi', 1008, 100)
('S002', 'Anu', 'Bangalore', 1010, 200)
('S003', 'Shahid', 'Bangalore', 1008, 175)
('S004', 'Ahila', 'Hydrabad', 1005, 235)
('S005', 'Girish', 'Hydrabad', 1003, 25)
('S006', 'Shylaja', 'Chennai', 1008, 180)
('S007', 'Lavanya', 'Mumbai', 1005, 325)
```

CHAPTER 16: Data visualization using pyplot: line chart, pie chart and bar chart

Choose the best answer: (1 Mark)

1. Which is a python package used for 2D charts?
   a) matplotlib.pyplot b) matplotlib.pip
   c) matplotlib.numpy d) matplotlib.plt
2. Identify the package manager for installing Python packages, or modules.
   a) Matplotlib **b) PIP** c) plt.show() d) python package
3. Which of the following feature is used to represent data and information graphically?
   a) Data List b) Data Tuple c) Classes and Objects **d) Data Visualization**
4. ..... is a collection of resources assembled to create a single unified visual display.
   a) Interface **b) Dashboard** c) Objects d) Graphics
5. Which of the following module should be imported to visualize data and information in Python?
   a) csv b) getopt c) mysql **d) matplotlib**
6. ..... is a type of chart which displays information as a series of data points connected by straight line segments.
   a) Line chart b) Pie chart c) Bar chart d) All the above
7. Read the code: import matplotlib.pyplot as plt
   plt.plot(3,2)
   plt.show()
   Identify the output for the above coding.



Ans: c

8. Identify the right type of chart using the following hints.
   Hint 1: This chart is often used to visualize a trend in data over intervals of time.
   Hint 2: The line in this type of chart is often drawn chronologically.
   a) Line chart b) Bar chart c) Pie chart d) Scatter plot
9. Read the statements given below. Identify the right option from the following for pie chart.
   Statement A: To make a pie chart with Matplotlib, we can use the plt.pie() function.
   Statement B: The autopct parameter allows us to display the percentage value using the Python string formatting.
   a) Statement A is correct b) Statement B is correct
   c) Both the statements are correct d) Both the statements are wrong

Answer the following questions: (2 Marks)

1. **What is Data Visualization?**
   - Data Visualization is the graphical representation of information and data.
2. **List the general types of data visualization.**
   - Charts * Tables * Graphs
   - Maps * Infographics * Dashboards
3. **List the types of Visualizations in Matplotlib.**
   - Line plot * Scatter plot * Histogram
   - Box plot * Bar chart and * Pie chart
4. **How will you install Matplotlib?**
   - **Matplotlib** can be installed using pip software.
   - Pip is a Package manager software for installing python packages.
5. **Write the difference between the following functions: plt.plot([1,2,3,4]), plt. plot([1,2,3,4], [1,4,9,16]).**

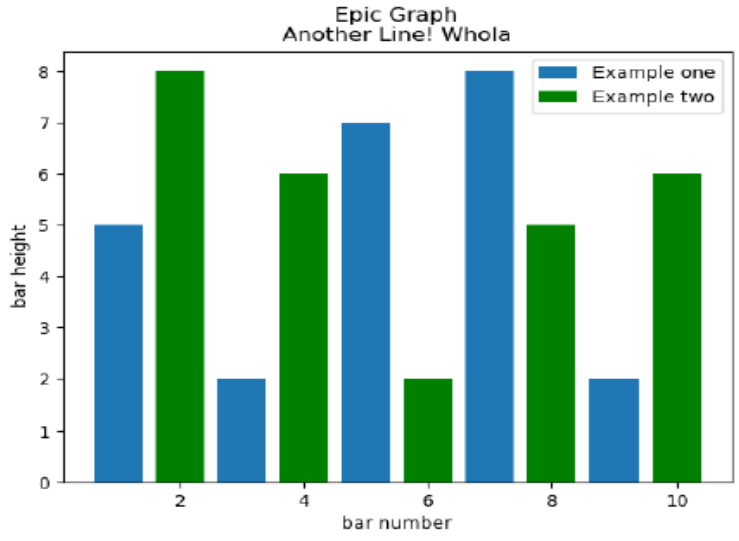
plt.plot([1,2,3,4])	plt.plot([1,2,3,4], [1,4,9,16])
It refers y value as [1,2,3,4]	It refers x and y values as ([1,2,3,4], [1,4,9,16])
Indirectly it refers x values as [0,1,2,3] (0,1) (1,1) (2,3) (3,4)	Directly x and y values are given as (1,1) (2,4) (3,9) (4,16)

Answer the following questions: (3 Marks)

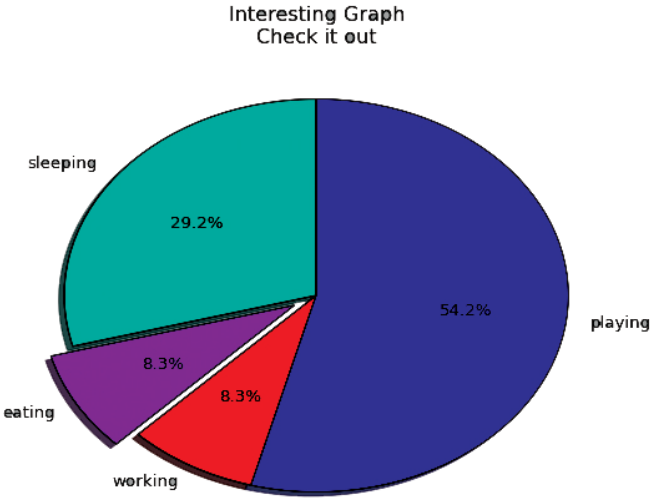
1. **Draw the output for the following data visualization plot.**

```
import matplotlib.pyplot as plt
plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Example one")
plt.bar([2,4,6,8,10],[8,6,2,5,6], label="Example two", color='g')
plt.legend()
plt.xlabel('bar number')
plt.ylabel('bar height')
plt.title('Epic Graph\nAnother Line! Whoa')
plt.show()
```

Output:



2. **Write any three uses of data visualization.**
   - Data Visualization help users to analyze and interpret the data easily.
   - It makes complex data understandable and usable.
   - Various Charts in Data Visualization helps to show relationship in the data for one or more variables.
3. **Write the plot for the following pie chart output.**



Coding:

```
import matplotlib.pyplot as plt
sizes=[29.2,8.3,8.3,54.2]
labels=["Sleeping","Eating","Working","Playing"]
cols=['c','m','r','b']
plt.pie(sizes,
labels=labels,
colors=cols,
startangle=90,
shadow=True,
explode=(0,0.1,0,0),
autopct='%1.1f%%')
plt.title('Intresting Graph\nCheck it out')
plt.show( )
```

Answer the following questions: (5 Marks)

1. **Explain in detail the types of pyplots using Matplotlib.**
   - **Line Chart:** A Line Chart is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments.
   - **Scatter plot:** A scatter plot is a type of plot that shows the data as a collection of points.
   - **Histogram:** Histogram refers to a graphical representation; that displays data by way of bars to show the frequency of numerical data.
   - **Box plot:** The box plot is a standardized way of displaying the distribution of data based on the five number summary: minimum, first quartile, median, third quartile, and maximum.
   - **Bar Chart:** Bar Chart shows the relationship between a numerical data and a categorical values.
   - **Pie Chart:** It is a circular graphic which is divided into slices to illustrate numerical proportion. The point of a pie chart is to show the relationship of parts out of a whole.
2. **Explain the various buttons in a matplotlib window.**
   - **Home Button** → If you ever want to return back to the original view, you can click on this.
   - **Forward/Back buttons** → These buttons can be used like the Forward and Back buttons in your browser.
   - **Pan Axis** → To click it, and then click and drag your graph around.
   - **Zoom** → The Zoom button lets you click on it, then click and drag a square that you would like to zoom into specifically. Zooming in will require a left click and drag. You can alternatively zoom out with a right click and drag.
   - **Configure Subplots** → This button allows you to configure various spacing options with your figure and plot.
   - **Save Figure** → This button will allow you to save your figure in various forms.
3. **Explain the purpose of the following functions:**

   a. plt.xlabel - specifies label for X-axis
   b. plt.ylabel - specifies label for Y-axis
   c. plt.title - specifies title to the graph
   d. plt.legend() - Calling legend() with no arguments automatically fetches the legend handles and their associated labels.
   e. plt.show() - Display a figure. When running in Python with its Pylab mode, display all figures and return to the Python prompt.

"Education Is Not Just About
Going To School And
Getting A Degree.
It's About
Widening Your Knowledge And
Absorbing The Truth About Life."

ALL THE BEST!

J. KAVITHA, B.Sc, B.Ed, M.C.A, M.Phil.,
Computer Instructor Gr - I
GHSS, S.S.KULAM
Coimbatore – 641107.
✆: 8940762362