Exam solution:

Q1:

1)

i) A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements. Each element or value that is inside of a list is called an item. Just as strings are defined as characters between quotes, lists are defined by having values between square brackets.

ii) Tuple is a collection of objects separated by commas. In some ways, a tuple is similar to a list in terms of indexing, nested objects, and repetition but a tuple is immutable, unlike lists which are mutable.

iii) The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable.

2)

i) The break statement provides you with the opportunity to exit out of a loop when an external condition is triggered. You'll put the break statement within the block of code under your loop statement, usually after a conditional if statement.

ii) The continue keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration.

iii) The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed. Empty code is not allowed in loops, function definitions, class definitions, or in if statements.

3) The self is used to represent the instance of the class. With this keyword, you can access the attributes and methods of the class in python. It binds the attributes with the given arguments. The reason why we use self is that Python does not use the '@' syntax to refer to instance attributes.

4) A Python docstring is a string used to document a Python module, class, function or method, so programmers can understand what it does without having to read the details of the implementation. Also, it is a common practice to generate online (html) documentation automatically from docstrings.

Q2:

1) False

2) True

3) False

4) True

5) True

Q3:

1) string=raw\_input("Enter string:")

vowels=0

for i in string:

if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):

vowels=vowels+1

print("Number of vowels are:")

print(vowels)

2) def recur\_sum(n):

if n <= 1:

return n

else:

return n + recur\_sum(n-1)

num = 16

if num < 0:

print("Enter a positive number")

else:

print("The sum is",recur\_sum(num))

3) nterms = int(input("How many terms? "))

n1, n2 = 0, 1

count = 0

if nterms <= 0:

print("Please enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence upto",nterms,":")

print(n1)

else:

print("Fibonacci sequence:")

while count < nterms:

print(n1)

nth = n1 + n2

# update values

n1 = n2

n2 = nth

count += 1

4) sample\_dict = {'a': 100, 'b': 200, 'c': 300}

if 200 in sample\_dict.values():

print('200 present in a dict')

5)

Q4: 1)

import random

import string

symbol=('!','@','#','-','@','+','%')

x= random.choice(symbol)

digit= random.randint(0,9)

if digit!=5:

y= digit

letter= random.choice(string.ascii\_letters)

if letter!= 'e' or 'E':

z= letter

def main():

list= x + str(y) + z

size= random.randint(6, 10)

pw=''.join(random.choice(list) for i in range(size))

print pw

main()

2)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport"

content="width=device-width, initial-scale=1.0" />

<title>Document</title>

<style>

body {

padding: 25px;

background-color: white;

color: black;

font-size: 25px;

}

.dark-mode {

background-color: black;

color: white;

}

.light-mode {

background-color: white;

color: black;

}

</style>

</head>

<body>

<h1>DarkMode</h1>

</p>

<h3 id="DarkModetext">Dark Mode is OFF</h3>

<button onclick="darkMode()">Darkmode</button>

<button onclick="lightMode()">LightMode</button>

<script>

function darkMode() {

var element = document.body;

var content = document.getElementById("DarkModetext");

element.className = "dark-mode";

content.innerText = "Dark Mode is ON";

}

function lightMode() {

var element = document.body;

var content = document.getElementById("DarkModetext");

element.className = "light-mode";

content.innerText = "Dark Mode is OFF";

}

</script>

</body>

</html>

Q5:

1) The default constructor is a type of constructor that is called automatically when the programmer has not defined any constructor in the program. In contrast, the parameterized constructor is a type of constructor defined by the programmer by passing parameters to give initial values to the instance variables in the class. That is the main difference between default and parameterized constructor.

2) Class is a detailed description, the definition, and the template of what an object will be. But it is not the object itself. Also, what we call, a class is the building block that leads to Object-Oriented Programming. It is a user-defined data type, that holds its own data members and member functions, which can be accessed and used by creating an instance of that class. It is the blueprint of any object. Once we have written a class and defined it, we can use it to create as many objects based on that class as we want. In Java, the class contains fields, constructors, and methods. For example, consider the Class of Accounts. There may be many accounts with different names and types, but all of them will share some common properties, as all of them will have some common attributes like balance, account holder name, etc. So here, the Account is the class.

Object is an instance of a class. All data members and member functions of the class can be accessed with the help of objects. When a class is defined, no memory is allocated, but memory is allocated when it is instantiated (i.e. an object is created). For Example, considering the objects for the class Account are SBI Account, ICICI account, etc.