Name	**Miandad Khan**
ID	**14130**
Semester	**6Th**
Subject	**OOPS**

Each question carry equal marks.   Please answer briefly.

Q1. What is Class and role of object in a Class, explain in detail with the help of a suitable program?

Ans:-

Class:-

A class can be defined as a template/blueprint that describes the behavior/state that object of its type support.

Example

```
public class Dog {
   String breed;
   int age;
   String color;

   void barking() {
   }

   void hungry() {
   }

   void sleeping() {
   }
}
```

A class can contain any of the following variable types.

- **Local variables** − Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.
- **Instance variables** − Instance variables are variables within a class but outside any method. These variables are initialized when the class is instantiated. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.
- **Class variables** − Class variables are variables declared within a class, outside any method, with the static keyword.

A class can have any number of methods to access the value of various kinds of methods. In the above example, barking(), hungry() and sleeping() are methods.

Following are some of the important topics that need to be discussed when looking into classes of the Java Language

Object:-

As mentioned previously, a class provides the blueprints for objects. So basically, an object is created from a class. In Java, the new keyword is used to create new objects.

There are three steps when creating an object from a class −

- **Declaration** − A variable declaration with a variable name with an object type.
- **Instantiation** − The 'new' keyword is used to create the object.
- **Initialization** − The 'new' keyword is followed by a call to a constructor. This call initializes the new object.

Following is an example of creating an object −

Example

Live Demo

```
Public class Puppy {
   Public Puppy(String name) {
      //This constructor has one parameter, name.
      System.out.println("Passed Name is :" + name );
   }

   Public static void main(String []args) {
      // Following statement would create an object myPuppy
      Puppy myPuppy = new Puppy( "tommy" );
   }
}
```

If we compile and run the above program, then it will produce the following result −

Output

```
Passed Name is:tommy
```

Q2. Write a program about table printing which takes input from the user on the basis of OOP and explain in detail?

Ans:-

Programs for tables Printings:-
Solution:-

num = int(input("enter the number: "))

//here we taking the number from the user

for i in range(1,(num +1)):

//now we creating "forloop" for i=1;

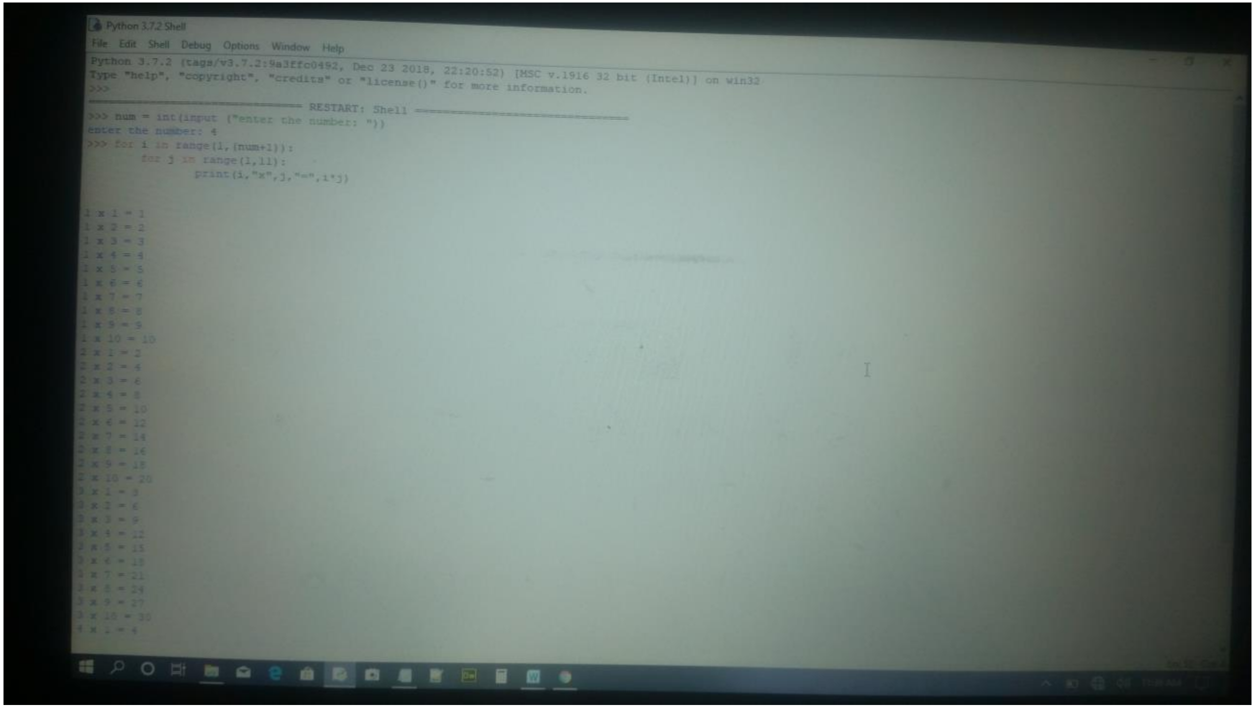
for j in range (1,11)

//now we creating "forloop" for j means 1to 10

Print (i, "x", j,"=", i+j)

//Here for the first time print the value of i=1 and j=1:
//then go the value of "i" from 1 upto 10
//after the 1upto 10 the "i" become 2 and then 3, 4 accordingly

So this is the whole process for printing the tables



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==================================== RESTART: Shell ==============================
>>> num = int(input ("enter the number: "))
enter the number: 4
>>> for i in range(1,(num+1)):
        for j in range(1,11):
                print(i,"x",j,"=",i*j)


1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
4 x 1 = 4
```

Q3. Write a program about any 2 cars which can calculate the performance of both of them and explain in detail?
Program:-

Example

```
import java.io.*;
public class Cars {

   int speed
   color;
   full speed;

   Overall Performance;

   // This is the constructor of the class Employee
   public Employee(String name) {
      this.name = name;
   }

   // Assign the speed of the Employee  to the variable speed.
   public void empspeed(int empspeed) {
      age = empspeed;
   }

   /* Assign color to both.*/
   public void empcolor(String empcolr) {
      color = empcolr;
   }

   /* Assign the full spee.*/
   public void empfull speed(double fullspeed) {
      full speed = full speed;
   }

   /* Print the Employee details */
   public void both calculation() {
      System.out.println("Both Car Performance:"+ int );
      System.out.println("color:" + colr );
      System.out.println("full speed:" + full speed );
      System.out.println("Overall Performance:" + Best);
   }
}
```

As mentioned previously in this tutorial, processing starts from the main method. Therefore, in order for us to run the both performance of the Car there should be a main method and objects should be created. We will be creating a separate class for these tasks.

Following is the performance of the Car class, which creates two instances of the class Employee and invokes the methods for each object to assign values for each variable.

Save the following code in EmployeeTest.java file.

```
import java.io.*;
public class EmployeeTest {

   public static void main(String args[]) {
      /* Create two objects using constructor */
      Employee empOne = new Cars("Black");
      Employee empTwo = new Employee("White");

      // Invoking methods for each object created
      empOne.speed(100);
      empOne.Color("Black");
      empOne.fullspeed(280);
      empOne.printCars();

      empTwo.speed(100);
      empTwo.color("white");
      empTwo.Full speed(290);
      empTwo.printCars();
   }
}
```

Now, compile both the classes and then run *cars perfprmance* to see the result as follows −

Output

```
color:black
speed:100
full Speed:280
overall Performnce: Good
color:white
speed:100
Full speed:280
overall Performnce: Best
```