Name Bilal Ahmad

ID 15089

Section B

Subject object-oriented-programming.

Question # 1

Answer:-

Before you create objects in Java, you need to define class.
A class is a blueprint for the object.
We can think of the class as a sketch (prototype) of a house. It contains all the details about the floors, doors, windows etc. Based on these description we build the house. House is an object. Since many houses can be made from the same description we can create many objects from a class.

For example:-

```
class className {
    // variables
    // methods
}
```

For example :-

```
Class Lamp {

    // instance variable private
     Boolean isON;

    // method public void
        turnOn(){
          isON=true;
    }
    // method
     Publice void turnoff(){
          isON = false;
     }
}
```

Here we have created class named Lamp.
The class has on variable named (isON). And two methods turnOn() & turnoff(). These variable and methods defined within class are called members of class.

In the example public & private know as access modifiers.

Objects in class:-

An objects are called an instance of a class.
For example, suppose [animal] is a class then Cat, Dog, Horse and so on can be considered as object of [Animal] class.

```
ClassName object = new classname();
```

Here we must using the Constructor ClassName() to create the object. Constructors have the same name as the class and are similar to methods.

For example:-

```
// L1 object
Lamp L1 = new Lamp();
// L2 object
Lamp L2 = new Lamp();
```

We have created objects named L1 and L2 using the constructor of Lamp class (Lamp()).

objects are used to access members of a class.

For example=

```
class Lamp

 void turnon() {
  ison =true;
 }

 }
  class classobject Example{
 Public Static void main {
 (String[] args) {
  L1 turnon();
}
}

  void turno77() {
  // initialize variable with value.
   ison = false;
   system.out.println("Light on?" +
}
  class main{
 Public Static void main (String[]args)
  .
  // Create objects L1 & L2
  Lamp L1 = new Lamp ();
  Lamp L2 = new Lamp ();

 // call method turnon() & turno77()
 L1. turnon();
 L2. turno77();
}

}
```

Question # 2:-

Answer:-

```
import java.utile.scanner;

public class work {

 Public static vaid main (string [] arg) {

    Scanner in = new scanner (system.in);
     int num1 = in.next Int();

    for (int i=0; i<10; i++){
     System.out.println (num1+"X"+
     (i+1)+"="+
      (num1*(i+1)));
    }
  }
}
```

Basics of OOP:-

object-oreinted programming has four basic concept: Encapsulation, Abstraction, inheritence and polymorphism Even if these concepts seems incredibly complex, understanding the general framework of how they work will help you understand the basics of computer program.

Enscapsulation -
Abstraction -
Inheritence -
Polymorphism -

→ Encapsulation :-

The different objects inside of one program will try to communicate with each other automatically. If a programmer wants to stop objects from interacting with each other, they need to be encapsulate in individual classes. Through the process of encapsulation classes cannot change or interact with the specific variables and functions of an object

Just like a pill "encapsulation" or contains the medication inside of its coating the principle of encapsulation works in a digital way to form a protective barrier around the information that separate it form the rest of the code. Programmers can replicate this object throughout different parts of the program or other program.

→ **Abstraction :-**

Abstraction is like an extension of encapsulation because it hides certain properties and methods from the outsides code to make the interface of the object simpler. Programmer use abstraction for several beneficial reasons. overall abstraction helps isolate the impact of changes made to the code so that if somethings goes wrong, the change will only affect the variable shown and not the outside code.

Inheritence :-

Programmers can extend the functionality of the code is existing classes of eliminate repetative code. For instance, element of HTML code that include a text box, select field and checkbox have certain properties in common with specific methods. Instead of redefining the properties and methods for each type of HTML element you can define them once in a generic object. Naming that object something like "HTML elements" will cause other objects to inherit its properties and methods so you can reduce unnecessary code.

→ Polymorphism :-

This techinques meaning "many forms or shapes". allows programmers to render multiple HTML. elements depending on the type of object. This concept allows programmers to redefine the way something works by changing how it is done or by changing the parts in which it is done. Terms of polymorphism are called overriding and overloading.

xxx —— xxx —— xxx —— xxx

Question # 3

Answer:

```
Public class cars
{
  Public static void main (string[] args)
  {
    Car car1 = new car();
    Car car2 = new car();

  System.out.println("This car is" + car1.
  get brand() + ", year" + car1.get year() +"
  Price" + car1.getprice());

  System.out.println("This car is" + car2.
  get brand() + ", year" + car2.get year() +"
  Price" + car2.get price();

  System.out.println("The total car No is:"
  + car1.getNo();
  System.out.println("The total car No is:"
  + car2.get No();
  }
{
```

You have created parameterized constructor i.e public Car(String, int number).
So when you are trying to create object for the same like Car car1 = new Car(); then it won't be possible Because in this you are trying to call default constructor - which is not present in the class.

End of Paper