Q # 01

ANSWER:-

Before you Creat objects in Java, you need to define class.
A class is a blueprint for the object. we can think of the class as a skatch (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 creat 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
    Public 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 acess modifiers.

OBJECT 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.

```
Class Name Object = new classname();
```

Here we must using the Constructor classname() to Creat 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

ID-15226 Date: #03

Class (Lamp ()).
objects area 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 turnoff () {
      // initialize variable with value.
       isON = false;
       System. old. print ln ("Light on}" + }
    }
      Class main {
    Public static void main (string [] args).
       // Creat objects L1 & L2
        Lamp L1 = new Lamp ();
        Lamp L2 = new Lamp ();

       // call method humon () & turnoff ()
        L1 turon ();
        L2 turnoff ();
    }
}
```

ID. 15226 #04

Q # 02

ANSWER:

```
Import java . utile . Scanner;

Public class work {

Public static void main (string [] arg)

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

 For (int i = 0; i < 10; i++) {
 System. out. print ln (num1 + " x " + (i+1) + "
   (num1 * (i+1) ");
    }
   }
}
```

ID.15226 #05 Date:

Enscapsulation.
Abstraction.
Inheritence.
Polymorplism -

ENCAPSULATION:

The different objects inside of one program will 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 off encapsulation classes cannot charge or intract with the specific variables and of an object.

Just like a pill "encapsulation" or contains the medication inside of encapsulation works in a digital way to from a protective barrier around the information that seperate it from the rest of the code. programmers can replicate this object throug out 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 outside code to make the interface of the object simpler programmer use abstruction for several beneficial reasons overall abstraction helps isolate the impact of changes made to the code so that if something goes wrong, the change will only affect the veriable shown and not the outside code.

ID.15226 #06

INHERITENCE

Programmers can extend the functionality of the codes existing classes of elimm-ate repetative code for instance, element of HTML code that include a text book, select field and checkbox have certain properties in common with specific methods instead of readfinnig the properties and methods for each type of HTML element you can define them once in a generic object. Nomming that object something like "HTML elevent-tive. will cause other objects to inherit its properties and methods so you can reduce unnecessary code.

POLYMORPHISM

This techiniques meaning "many froms or shapes."
allows programmers to render multiple HTML. elenests depending on the type of object. This concept allows progra-mmers 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 everloading.

Q # 03

ANSWER

```
Import Java.util.scanner;
Public Class Exercise 12 {
```

ID · 15226 #07

```
Public static void main(string[] args) {
Scanner in = new scanner (system.in);

System.out.print("Input Speed(KM/H) of first car:
");
Int car1 = in.nextInt();

System.out.print("input Speed(KM/H) of Second
Car:");
Int car2 = in.nextInt();

System.out.println("performance of tow cars
is:" +
(car1 + Car2)/2);
  }
}
```

Date: _____

NAME. Shabir Ahmad

ID . 15226