*Dt : 20/8/2022*

*(a)Class:*

*=>class is a 'Structured layout' in java and which generate objects.*

*=>Class is a collection of 'Variables and methods'.*

*=>Classes in Java are categorized into two types:*

*1.Pre-defined classes*

*2.User defined classes*

*1.Pre-defined classes:*

*=>The classes which are already defined and available from JavaLib are known as Pre-defined Classes or Built-in classes.*

*Ex:*

*String*

*System*

*2.User defined classes:*

*=>The classes which are defined by the programmer are known as User defined classes or Custom classes.*

*Ex:*

*Display.java*

*Addition.java*

*==========*

**imp*

*Variables in Java:*

*=>Variables are data holders in programs.*

*=>based on datatype the variables are categorized into two types:*

*1.Primitive DataType variables*

*2.Non-Primitive DataType variables*

*1.Primitive DataType variables:*

*=>The variables which are declared with primitive datatypes like byte,short,int,long,float,double,char,boolean are known as Primitive DataType variables.*

*=>These primitive datatype variables will hold values.*

*2.Non-Primitive DataType variables:*

*=>The variables which are declared with Non-Primitive datatypes like 'class,interface,Array,Enum' are known as Non-Primitive DataType variables.*

*=>These Non-Primitive DataType variables will hold Object references.*

*==========*

**imp*

*=>Based on 'static' keyword the variables are categorized into two types:*

*1.Static variables*

*2.Non-static variables*

*1.Static variables:*

*=>The variables which are declared with 'static' keyword are known as static variables or Class variables.*

*=>These static variables will get the memory within the class while Class_loading.*

*=>These static variables canbe accessed with Class_name.*

*2.Non-static variables:*

*=>The variables which are declared without 'static' keyword are known as Non-Static variables.*

*=>These Non-Static variables are categorized into two types:*

*(a)Instance Variables*

*(b)Local Variables*

*(a)Instance Variables:*

*=>The Non-Static variables which are declared outside the methods are known as Instance variables or Object variables.*

*=>These Instance variables will get the memory within the object while object creation.*

*=>These Instance variables can be accessed with Object_name.*

*(b)Local Variables:*

*=>The Non-Static variables which are declared inside the methods are known as Local Variables or Method Variables.*

*=>These Local variables will get the memory within the method while method execution.*

*==========*

**imp*

*Methods in Java:*

*=>Methods are the actions which are executed to generate result.*

*=>Methods in Java are categorized into two types:*

*1.Static methods*

*2.Non-Static methods or Instance methods*

*1.Static methods:*

*=>The methods which are declared with 'static' keyword are known as static methods or Class methods.*

*=>These static methods will get the memory within the class while class loading.*

*=>These static methods canbe accessed with Class_name.*

*Structure of static methods:*

```
static return_type method_name(para_list)
{
 //method_body
}
```

***Coding Rule:***

***=>Static methods can access static variables directly,but cannot access Instance variables directly.***

***2.Non-Static methods or Instance methods:***

***=>The methods which are declared without static keyword are known as NonStatic methods or Instance methods***

***=>These Instance methods will get the memory within the object while object creation.***

***=>These Instance methods canbe accessed with Object_name.***

***Structure of Instance methods:***

```
return_type method_name(para_list)
{
 //method_body
}
```

***Coding Rule:***

***=>Instance methods can access both static variables and Instance variables.***

***==========***

***Ex : DemoVariables.java***

```
class  DemoVariables
```

```
{
        static int a=10;
        int b=20;
        static void m1()
  {
    System.out.println("====static m1()====");
                System.out.println("The value of a="+a);
    //System.out.println("The value of b="+b);

  }
        void m2()
  {
                System.out.println("====Instance m2()=====");
                System.out.println("The value of a="+a);
                System.out.println("The value of b="+b);
  }
        public static void main(String[] args)
        {
                int c=30;
                DemoVariables.m1();
                DemoVariables ob = new DemoVariables();
                ob.m2();
        }
}
```

***o/p:***

***====static m1()====***

***The value of a=10***

***====Instance m2()=====***

***The value of a=10***

***The value of b=20***

***==========***