# Java Test 3

Q1: Write a program to check given number is prime or not.

A1:

```
import java.util.*;
class PrimeNumber
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter a number : ");
                int n=sc.nextInt();
                int counter=0;
                for( int i=2; i<=n/2; i++)
                {
                        If(n%i==0)
                        {
                                counter=1;
                                break;
                        }
                }
                If(counter==0)
                        System.out.println("Number is Prime. ");
                else
                        System.out.println("Number is Prime. ");
        }
}
```

Q2: What is type casting? Give Example.

A2:

## Type Casting

The conversion of data from one data type to another is called type conversion (or) type casting. This is classified in two types.

1.     Implicit type casting.
2.     Explicit type casting.

1. **Implicit type casting:**

   Implicitly Java provides type casting. Java automatically promotes values to a higher data type; to prevent any loss of information. Consider the following example.

   > byte b=100;
   > int a=b;

   Here the byte value is automatically promoted to higher data type int and then, the value is stored. This is done by java automatically, so it is called automatic type conversion (or) **implicit type conversion**. Here there is no data loss. Java allows, you to make certain assignment conversion by assigning the value of a variable of one type to another without an explicit cast.

   i.  byte —> short—>int —> long —> float —> double.
   ii. char —> int —> long -> float —>double

   Where you can assign a variable of type that is to the left to the on its right in the list above. It means from byte to a short, an int, a long, a float, or a double. This is also called widening conversion.

2. **Explicit Type Conversion**

   Consider the following statement int i=7.713; Here, we can assign higher data type (double) to lower data type (int), the data can be also lost. The factional component is lost. So we call this type of situation is "narrowing conversion", assign larger type to smaller one is called narrowing conversion. So we must explicitly declare the type casting.

General form:

E.g.: (type) expression

int i=(int) 7.713;

If you are not explicitly casting, the compiler give an error message, such as "Incompatible type for declaration. Explicit cast needed to convert double to int".


Q3: Write a program to solve following puzzle.  A frog starts climbing 30 ft well. Each hour frog climbs 3 ft and slips back 2 ft. How many hours does it take him to reach top and get out?
A3:

```
class puzzle
{
        Public static void main (String args[])
        {
                int wellhight=30;
                int climbsparhour=3;
                int slipsparhour=2;
                int hours=0;
                int climbs=0;
```

```java
        while (climbs< wellhight)
        {
                climbs= climbs + climbsparhour ;
                hours= hours+1;
        // which hour frog climbs value greater or equal then not slip down
                if(climbs< wellhight )
                {
                        climbs= climbs  – slipsparhour;
                }
        }
        System.out.println(“A frog starts climbing “+wellhight +” ft well”);
        System.out.println(“Each hour frog climbs “+climbsparhour +” ft”);
        System.out.println(“Each hour frog slips back “+slipsparhour +” ft”);
        System.out.println(hours +“hours does it take him to reach top and get out”);
    }
}
```

Output:

```
>java puzzle
A frog starts climbing  30 ft well
Each hour frog climbs  3 ft
Each hour frog slips back  2 ft
28hours does it take him to reach top and get out
```

Q4: What is an object and class? Explain with an example.

A4: **Class:**  A class is usually described as the blueprint or template from which, the object is actually made. Its attributes and methods represent a class. A class is described by use of the class keyword. The attributes and methods (behavior) of a class are defined inside a class body. In java, the braces {} mark the beginning and the end of a class or method.
Here data, or variable defined with in a class are called in stance variable. The code is contained within a method. These are called as members of the class. A class name is mandatory and must be given while declaring a class. This name is used to refer the class and to create instance of the class. The class keyword is used to declare a class. The optional part access specifier and modifier

**Object:**   An object is an instance of the class. A software object maintains its states in variables and implements its behavior with methods. Declaring on object is similar to declaring a variable. You must allocate memory to object before you use the object. This can be achieved by new operator.

```java
        sony= new TV();//create
        //(or) simply done by same line
        TV sony =new TV(); //declare and creation
```

Q5: What are the difference between constructor and a method?

A5:

**Difference between a Constructor and a Method**

| Constructor | Method |
|---|---|
| It has the same name as the class itself. | It has its own name, because it is an ordinary member function of a class. |
| Constructor has no return type. | Method has return type (which may be void). |
| Constructors are invoked by the new operator. | Method is invoked using the dot operator. |
| Syntax<br>[modifier]<class name>(parameter list)<br>{<br>//body of the statements;<br>} | Syntax<br>[access specifier] [modifier]<return-type><br><method name> (parameter list)<br>{<br>//body of the statements;<br>} |