```
1.class DataType
{
  public static void main(String[] args)
   {
           byte a=112;
           short b=32765;
           int c=1234567898;
           //int c=2234567890; integer number is too large>(2^31)-1 so used long
type.
           long d=2234567890L;
           //float f=15.50;possible lossy conversion from double to float
           float f=15.50f;
           double g=154.50;
    System.out.println("@");   //@ as a string.
        System.out.println('@');   //@ as a character.
        System.out.println('@'+10);//64+10=74.
        System.out.println('@'+"10");//concatenation.
        System.out.println('\u0001');
        System.out.println('\u0011');
        System.out.println('\u0123');
        System.out.println('\u0011'+'\u0123');
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(d);
        System.out.println(f);
        System.out.println(g);
        System.out.println("a");   //@ as a string.
        System.out.println('a');   //@ as a character.
        System.out.println("a"+20);//concatenation.
        System.out.println('a'+20);//97(ASCII VALUE OF a)+20=117.
        System.out.println('a'+"20");//concatenation.
        System.out.println(a+20);//variable and operation(112+20=132).


        }
}
```

```
2.class Test
 {
    static int a=10;//global variable(static).
               int b=20;//global variable(Non static).
        public static void main(String[] args)
          {
```

```
            int c=30;//local variable.
                {
                System.out.println(c);
                System.out.println(a);//static variable can call directly.
                System.out.println(Test.a);//or static variable is called  by
giving class name as reference.
                Test t1=new Test(); //object creation for non static variable.
        System.out.println(t1.b);//call by object reference name.
            }
          int d=40;
          System.out.println(d);
          System.out.println(c+d);
          }
 }


3.class Variable
{
    public static void main(String[] args)
       {
             int a=12113;
                 int[] c={20,38,503,408};
                 String[] p={"Mohan","Sohan"};
                 int[] x={12,15,18};
                 String[] y={"12","Ram","28.56"};
                 System.out.println(a);//variable value.
                 System.out.println("a");//as String.
                 System.out.println('a');//as Character.
                 System.out.println('a'+10);//ASCII value+10(97+10=107).
                 System.out.println(a+10);//variable value+10(12113+10=12123);
                 System.out.println(c[0]);//20 as placed at 0 position.
                 System.out.println(c[2]);//503 as placed at 2nd position.
                 //System.out.println(c[4]);Exception:ArrayIndexOutOfBounds.
                 System.out.println(c[1]+c[3]);//38+408=446.

                 System.out.println(p[1]);//Sohan as placed at 1 position.
                 System.out.println(p[0]+p[1]);//concatenation.
                 System.out.println(y[0]+y[1]);//concatenation.
                 System.out.println(y[1]+x[0]+y[2]);//concatenation.
                 System.out.println(x[0]+x[2]);//add opearation.
                 System.out.println(x[0]+x[2]+y[1]+x[1]+x[2]);//addition and then
concatenation.

        }
}

4.class VariableProperty
{
        static int a;
        static double b;
```

```
        static String z;
    int c;//Non static variable

                public static void main(String[] args)
         {
                        {
                        int d;
                        int e=30;
                        int f=60;
                        //int e=40;error already e is initialized
                        System.out.println(a);//o/p as default value of integer=0.
                        System.out.println(b);//o/p as default value of double=0.0
                        System.out.println(z);//o/p as default value of
String=null.
                        VariableProperty v1=new VariableProperty();
                        System.out.println(v1.c);
                        //System.out.println(d);Error as d is not initialized.local
variable must be initialized.
                        a=10;
                        System.out.println(a);//can update the value of the
variable.
                        a=45;
                        System.out.println(a);
           }
                   {
                         int e=40;
                         System.out.println(e);//same variable can be initialized
outside the block.
                         //System.out.println(f);//a variable is accessible only
inside the same block.
                    }
                    System.out.println(a);
                  }
}

5.class Concatenation
{
        public static void main(String[] args)
        {
                  String a="Mohan";
                  String b="Sohan";
                  int x=35;
                  int y=40;
                  int z=x+y;
                  int w=x*y;
                  System.out.println(a+b);
                  System.out.println(a+b+48);
                  System.out.println ("The Sum of x & y is:"+z);
                  System.out.println("The sum of "+x+" and "+y+" is:"+z);
                  System.out.println("The multiplication of"+" "+x+" "+"and"+"
```

```
"+y+" "+"is:"+w);
                    System.out.println(40+35+" is the value");//75 then concatenation
with string.
                    System.out.println(40+35+" is the value "+30+52);//75 then
concatenate with string and all things after string.
                    //System.out.println("The value is:"+40-22);error (The value
is40)-20 data type error.
                    System.out.println(30-25+"is the value");//operation then
concatenation.
                    System.out.println("the value is "+(40+38));//follow BODMAS then
concatenation.
                    System.out.println("the value is "+(40*8));//follow BODMAS then
concatenation.
                    System.out.println("the value is "+(40/7));//follow BODMAS then
concatenation.
                    System.out.println("the value is:"+30+(60*8)+(30-5));//follow
BODMAS Then concatenation.

                }
}

6.class Casting
  {
   public static void main(String[] args)
     {
          int a=15;
          double b=(double)a;
          char x='@';
          int y=x;
          System.out.println(a);
          System.out.println(b);//Widening casting.
      System.out.println(x);
          System.out.println(y);//Widening casting print ASCII value of @.
          double d=15.523;
          //int e=d;Casting operator required.
          int e=(int)d;
          System.out.println(d);
          System.out.println(e);//Convert double into int by casting operator.
      int p=100;
      char q=(char)p;
          System.out.println(p);
          System.out.println(q);//print ASCII value of p(64).
         }
  }
7.class Operator
{
  public static void main(String[] args)
   {
     int a=38;
         int b=5;
```

```
         int c=a/b;//Arithmetic operator.
         double d=a/b;//Arithmetic operator.
         double x=48;
         int y=7;
         double z=x/y;//Arithmetic operator.
         //int w=x/y;Narrowing casting so Casting operator required.
         int w=(int)x/y;
        System.out.println(c);//(7)
        System.out.println(d);//Widening casting.(7.0)
        System.out.println(z);//Widening casting.(6.857142857142857)
        System.out.println(w);//Narrowing.(6)

        int p=12;
        int q=p+30;//Assignment Operator.
        int r=p+q;//Assignment Operator.

        System.out.println(p);//(12)
        System.out.println(q);//(42)
        System.out.println(r);//(54)

        p+=20;//p=p+20,Compound assignment/shorthand operator.
        q*=10;//q=q*10,Compound assignment/shorthand operator.
        r%=5;//r=r%5,Compound assignment/shorthand operator.
        System.out.println(p);//p=p+20(12+20=32)
        System.out.println(q);//q=q*10(42*10=420)
        System.out.println(r);//r=r%5(54%5=4)


        System.out.println(p>q);//Relational operator,32>420-false.
        System.out.println((p+q)>p-q*2);//Relational operator,452>-808-true
        System.out.println((p+q)/4==0);//Relational operator,113==0-false.
        System.out.println((p+q)%4==0);//Relational operator,0==0-true.
        System.out.println((p+q)%4!=0);//Relational operator,0!=0-false.

        }
}

8.class Increment
{
   public static void main(String[] args)
   {
         {
           int a=12;
           int b=a++;
           b++;
           int c=a+++--b;
           System.out.println(a);
           System.out.println(b);
           System.out.println(c);
         }
```

```
        {
         int a=10;
         a++;
         int b=++a;
         int c=(a++)+(++b);
         b++;
         System.out.println(a);
         System.out.println(b);
         System.out.println(c);
        }
        {
    int x=12;
        int y=x++ + ++x;
        y++;
        int z=++y;
        int p=x++ - ++y + z++;
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
        System.out.println(p);
       }
   }
}


9.class LogicalOperator
{
  public static void main(String[] args)
  {
        {
          int a=12;
          int b=10;
          int c=18;
          int d=a++ + ++b + c++ - --a;
          System.out.println(d);
        }
        {
                int a=12;
                a=a++;
                a=a++;
                a=a++;
                a=a--;
                a=a++;
                System.out.println(a);
        }
        {
                int a=12;
                a=++a;
                a=a++;
                a=++a;
```

```
                System.out.println(a);
        }
        //Logical AND Operator(&& or &)
        {
                int a=12;
                int b=20;
                System.out.println(true&& 10>5);
                System.out.println(8>5& 10>5);
                System.out.println(a++>10&&b++>10);
                System.out.println(a);
                System.out.println(b);
        }
        {
                int a=12;
                int b=20;
                System.out.println(a++>12&&b++>10);//&&b++>10 not executed as 1st
is false so '&& is fast b/c skip the 2nd operand if 1st operand is false'.
                System.out.println(a);//13
                System.out.println(b);//20
        }
        {
            int a=12;
                int b=20;
                System.out.println(a++>12&b++>10);//&b++>10 executed as 1st is
false,'& is Slow b/c both operand executes.
                System.out.println(a);//13
                System.out.println(b);//21
        }
        //Logical OR Operator(||)
        {
                int a=12;
                int b=20;
                System.out.println(a++>10||b++>16);//||b++>16 not executed as 1st
is true so '|| is fast b/c skip the 2nd operand if 1st operand is true'.
                System.out.println(a);//13
                System.out.println(b);//20
        }
        {
                int a=12;
                int b=20;
                System.out.println(a++>12||b++>10);
                System.out.println(a);//13
                System.out.println(b);//21
        }
        {
                int a=12;
                int b=20;
                System.out.println(++a>12|b++>10);//|b++>16  executed as 1st is
true so '| is slow b/c execute both operand.
                System.out.println(a);//13
```

```
                System.out.println(b);//21
        }
    }
}
```