1. Write the output?

```
 public class M2
 {
     public static void main(String [] args)
     {
         Double d1 = new Double(10.09);
         double d2 = d1.doubleValue();
         System.out.println(d1);
         System.out.println(d2);
         System.out.println("done");
     }
 }
```

2. Write the output?

```
public class M5
{
 public static void main(String [] args)
 {
     String s1 = "10";
     Integer obj = new Integer(s1);
     int i = obj.intValue();
     System.out.println(obj);
     System.out.println(i);
     System.out.println("done");
 }
}
```

3. Write the output?

```
public class M6
{
 public static void main(String [] args)
 {
     String s1 = "9.989";
     Double d1 = new Double(s1);
     double d2 = d1.doubleValue();
     System.out.println(d1);
     System.out.println(d2);
     System.out.println("done");
 }
}
```

4. Write the output?

```
public class M7
{
   public static void main(String [] args)
   {
       String s1 = "10Z";
       Byte b1 = new Byte(s1);
       byte b2 = b1.byteValue();
       System.out.println("done");
   }
}
```

5. Write the output?

```
public class M8
{
    public static void main(String [] args)
    {
        String s1 = "true";
        String s2 = "false";
        String s3 = "10Z";
        Boolean b1 = new Boolean(s1);
        Boolean b2 = new Boolean(s3);
        Boolean b3 = new Boolean(s3);
        boolean b4 = b1.booleanValue();
        boolean b5 = b2.booleanValue();
        boolean b6 = b3.booleanValue();
        System.out.println(b4);
        System.out.println(b5);
        System.out.println(b6);
        System.out.println("done");
    }
}
```

6. Write the output?

```
public class M9
{
    public static void main(String [] args)
    {
        String s1 = "a";
```

```
            Character c1 = new Character(s1);
            char c2 = c1.charValue();
            System.out.println("done");
        }
    }
```

7. Write the output?

```
public class M18
{
    public static void main(String [] args)
    {
        String s1 = "abc";
        String s2 = "true";
        String s3 = "TRUE";
        boolean b1 = Boolean.parseBoolean(s1);
        boolean b2 = Boolean.parseBoolean(s2);
        boolean b3 = Boolean.parseBoolean(s3);
        System.out.println(b1);
        System.out.println(b2);
        System.out.println(b3);
    }
}
```

8. Assume we compile this program in JDK 1.4
   Write the output for the following program?

```
public class M23

{

 public static void main(String [] args)

 {

     int i = 10;

     int j = new Integer(10);//unboxing

     System.out.println("done");

 }

}
```

9. Write the output for the following program?

```
public class M25
{
 public static void main(String [] args)
 {
     Double d1 = new Double(20);
     double d2 = d1.doubleValue();
     System.out.println(d1);
     System.out.println(d2);
     System.out.println("done");
 }
}
```

10.

```
Assume
We are compiling and running in JDK 1.5
What is the output?
Where auto-boxing and unboxing happening?
public class M27
{
 static void test(Integer obj)
 {
     System.out.println("done");
 }
 public static void main(String[] args)
 {
     test(10);
     test(new Integer(10));
 }
}
```

11. Assume

```
We are compiling and running in JDK 1.5
What is the output?
Where auto-boxing and unboxing happening?
public class M28
{
 static void test(int i)
 {
     System.out.println("done");
 }
 public static void main(String [] args)
 {
     Integer obj = new Integer(10);
     test(obj);
 }
}
```

12. Assume

```
   We are compiling in JDK 1.4
   What is the result?
   public class M31
    {
        static void test(Character obj)
        {
            System.out.println("done");
        }
        public static void main(String [] args)
        {
            test('a');
            char c1 = 't';
            test(c1);
            Character c2 ='p';
            test(c2);
        }
    }
```

13. Assume

We are compiling and running in JDK 1.5
What is the output?
Where auto-boxing and unboxing happening?

```
public class M32
{
   public static void main(String [] args)
   {
       Boolean b1 = new Boolean(true);

       if(b1)
       {
           System.out.println("done");
       }
   }
}
```

14. Assume

We are compiling and running in JDK 1.5
What is the output?
Where auto-boxing and unboxing happening?

```
public class M34
{
   static int test()
   {
       Integer obj = new Integer(20);
       return obj;
```

```
    }
    public static void main(String [] args)
    {
        Integer o1 = test();
        System.out.println("done");
    }
}
```

15. Assume

We are compiling and running in JDK 1.4

What is the output?

```
public class M36
{
    static void test(Integer obj)
    {
        System.out.println("Integer");
    }
    public static void main(String [] args)
    {
        int i = 20;
        Integer obj = new Integer(20);
        test(i);
        test(obj);
    }
}
```

16. Assume

We are compiling and running in JDK 1.5

What is the output?

```
public class M38
{

   static void test(double d1)
   {
       System.out.println("double");
   }
   static void test(Integer obj)
   {
       System.out.println("Integer");
   }
   public static void main(String [] args)
   {
       int i = 20;
       test(i);
   }
}
```

17. Assume

We are compiling and running in JDK 1.5

What is the output?

```
 public class M42
{
   static void test(String...str)
```

```
    {
        System.out.println("var args");
    }
    public static void main(String [] args)
    {
        test();
        test("abc");
        test("msg1", "xyz", "hello");
    }
 }
```

18. Assume
   We are compiling and running in JDK 1.5
   What is the output?

```
public class M44
{
    static void test(String...varArg)
    {
        for(String s1 : varArg)
        {
            System.out.print(s1 + ",");
        }
        System.out.println();
    }
    public static void main(String [] args)
    {
        test("abc");
        test("hello","test");
        test("hello","test","xyz");
        test("hello","test","xyz","abc");
    }
}
```

19.

Assume

We are compiling and running in JDK 1.5

What is the output?

```
class Z11 extends Object
{
    int i;
    public String toString()
    {
        return "" + i;
    }
    public static void main(String[] args)
    {
        Z11 z1 = new Z11();
        z1.i = 10;
        System.out.println(z1);
    }
}
```

20. List out the wrapper classes available?
21. Why do we need wrapper classes?
22. What are the different types of derived typecasting?

23. Apart from constructor, is there any other way to create corresponding Wrapper object?

24. Which Wrapper class has only one constructor? And why?
25. In which version of JDK, autoboxing and unboxing concept has been introduced?
26. Tell something about Autoboxing.

27. Tell something about Unboxing.
28. Explain about vararg in java.