CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

Advanced Computing Training School

Core Java (25 Minutes)

1. What is the result of this program?

```
class Over
{
  public static void main(String[] args){
       Under u = new Under();
       u.test();
  }
  int test(){
       System.out.println("over");
       return 1;
  }
}
class Under extends Over{
  short test(){
       super.test();
       System.out.println("Under");
       return 1;
  }
}
```

1. This code compiles, runs and displays over followed by Under
2. This code compiles, runs and displays Under followed by over
3. This code does not compile
4. Code will compile but gives runtime error

Correct Answer: 3

2. Consider the following code in file Sample.java

```
public class Sample implements IInt
{
  public static void main(String[] args){
       Sample s = new Sample();   //1
       int j = s.thevalue;        //2
       int k = IInt.thevalue;     //3
       int l = thevalue;          //4
  }
}
interface IInt
{
  int thevalue = 0;
}
```

What will happen when the above code is compiled and run?

1. It will give an error at compile time at line //1
2. It will give an error at compile time at line //2
3. It will give an error at compile time at line //3
4. It will compile and run without any problem.

Correct Answer: 4

3. What will be the result of attempting to compile and run the following program?

```
public class TestClass
{
  public static void main(String args[ ] ){
       String s = "hello";
       StringBuffer sb = new StringBuffer("hello");
       sb.reverse();
     s.reverse();
     if( s = = sb.toString() ) System.out.println("Equal");
     else
     System.out.println("Not Equal");
   }
 }
```

1. It will print 'Equal'
2. It will print 'Not Equal'
3. Compilation error as there is no reverse () method in class String
4. Runtime error

Correct Answer: 3

4. What will be the output of the following code?

```
public class exception_demo
{
  public static void main(String str[]){
     int i=1, j=1;
     try
       {
            i++;
            j--;
            if(i/j > 1)
            i++;
       }
     catch(Exception e)
       { System.out.println("Exception"); }
     catch(ArithmeticException e)
       { System.out.println("arithmetic exception"); }
     catch(ArrayIndexOutOfBoundsException e)
       { System.out.println("Array index exception"); }
     finally
       {     System.out.println("finally"); }
     System.out.println("after exceptions ");
  }
}
```

1. Give compilation error
2. arithmetic exception
3. arithmetic exception finally
4. None of the above

Correct Answer: 1

6. Analyze the following code:
```
public class Test{
   int x;
   static {x++;}
}
```
   1. The program cannot be compiled, because the statement x++ must be placed inside a method or a constructor.
   2. When you construct an instance of Test, the value of x becomes 0.
   3. The program cannot be compiled, because x is non-static, but is used in a static initialization block.
   4. When you construct an instance of Test, the value of x becomes 1.

Correct Answer: 3

7. If you will run following code what will be the result?
```
public class RTExcept {
      public static void throwit () {
            System.out.print("throw it ");
            throw new RuntimeException();
      }
      public static void main(String [] args) {
            try {
                     System.out.print("hello ");
                     throwit();
                  }
      catch (Exception re ) {
            System.out.print("caught ");
      }
      finally {
            System.out.print("finally ");
      }
            System.out.println("after ");
      }
      }
```
   1. hello throw it caught finally after
   2. hello throw it RuntimeException caught after
   3. Compilation fails
   4. hello throw it caught finally after RuntimeException

Correct Answer: 1

8. Which collection class allows you to access its elements by associating a key with an element's value, and provides synchronization?
   1. java.util.SortedMap
   2. java.util.TreeMap
   3. java.util.TreeSet
   4. java.util.HashTable

Correct Answer: 4

9. Which one is true about interface and abstract class?
   1. Abstract class can have only instance method and default behavior. Interface can declare constants and can have instance method but cannot implements default behavior.
   2. An interface has all public members and abstract class has private, protected etc members
   3. Both 1 & 2
   4. None of the above

Correct Answer: 3

10. Objects are passed by value or reference?
    1. By value
    2. By reference
    3. It depends upon how you specify
    4. None of the above

Correct Answer: 1

11. If you write System.exit(0) at the end of try block, will the finally block still execute?
    1. Yes
    2. No
    3. It depends upon return statement
    4. Can't say

Correct Answer: 2

12. Which is a keyword?
    1. string
    2. unsigned
    3. Float
    4. this

Correct Answer: 4

13. Which is valid declaration of a String?
    1. String s2 = 'null';
    2. String s3 = (String) 'abc';
    3. String s1 = null;
    4. String s4 = (String) '\ufeed';

Correct Answer: 3

14. Which is valid declaration within an interface?
    1. public static short stop = 23
    2. protected short stop = 23
    3. transient short stop = 23;
    4. final void madness(short stop);

Correct Answer: 1

15.
```
class  Equals{
   public static void main(String[] args){
         int x= 100;
         double y = 100.1;
         Boolean b = (x=y);
         System.out.println(b);
   }
}
```
    1. true
    2. false
    3. Compilation fails
    4. An exception is thrown at runtime

Correct Answer: 3

16. Line 1. long test(int x, float y)
    Line 2. {
    Line 3.

Line 4. }
The above program will not compile by inserting which of the following line?

1. return x;
2. return (long) x/y
3. return(int) 3.14d
4. return (long)y;

Correct Answer: 2

17. Which statement is true about wrapper or String classes?
    1. if x and y refer to instances of different wrapper classes, then the fragment x.equals(y) will cause a compiler failure.
    2. if x and y refer to instances of different wrapper classes, then x==y can sometimes be true.
    3. If x and y are String references and if x.equals(y) is true, then x==y is true.
    4. If x,y and z refer to instances of wrapper classes and x.equals(y) is true, and y.equals(z) is true, then z.equals(x) will always be true.

Correct Answer: 4

18.
```
String x = "xyz";
x.toUpperCase();
String y = x.replace('Y','y');
y = y + "abc" ;
System.out.println(y); What is the result?
```
1. abcXyz
2. abcxyz
3. xyzabc
4. compilation fails

Correct Answer: 3

19.
```
String a = "newspaper";
a = a + b;
char b = a.charAt(1);
a = a + b;
System.out.println(a); What is the result?
```
1. apa
2. app
3. apea
4. apep

Correct Answer: 2

20.
```
public class SqrtExample{
   public static void main(String [] args){
         double value = -9.0;
         System.out.println(Math.sqrt(value));
   }
}
```
1. 3.0
2. –3.0
3. NaN
4. Compilation fails

Correct Answer: 3