

1.

```
(1) public interface Foo {  
(2)     int k = 4;    /* Line 2 */  
(3) }
```

Which three piece of codes are equivalent to line 2?

- 1) final int k = 4;
- 2) public int k = 4;
- 3) static int k = 4;
- 4) abstract int k = 4;
- 5) volatile int k = 4;
- 6) protected int k = 4;

- A.. 1, 2 and 3
- B. 2, 3 and 4
- C. 3, 4 and 5
- D. 4, 5 and 6

2. What will be the output of the program?

```
public class Test107 implements Runnable {  
    private int x;  
    private int y;  
  
    public static void main (String args[]) {  
        Test107 that = new Test107();  
        (new Thread(that)).start();  
        (new Thread(that)).start();  
    }  
  
    public synchronized void run () {  
        for (int i = 0; i < 10; i++) {  
            x++;  
            y++;  
            System.out.println( "x = " + x + ", y = " + y ); /* Line 15 */  
        }  
    }  
}
```

- A. Compilation error.

B. Will print in this order: x = 1 y = 1 x = 2 y = 2 x = 3 y = 3 x = 4 y = 4 x = 5 y = 5... but the output will be produced by both threads running simultaneously.

C.. Will print in this order: x = 1 y = 1 x = 2 y = 2 x = 3 y = 3 x = 4 y = 4 x = 5 y = 5... but the output will be produced by first one thread then the other. This is guaranteed by the synchronised code.

D. Will print in this order x = 1 y = 2 x = 3 y = 4 x = 5 y = 6 x = 7 y = 8...

3.

```
Class Test1 {
    public int value;
    public int hashCode() { return 42; }
}
class Test2 {
    public int value;
    public int hashCode() { return (int)(value^5); }
}
```

which statement is true?

A. class Test1 will not compile.

B. The Test1 hashCode() method is more efficient than the Test2 hashCode() method.

C.. The Test1 hashCode() method is less efficient than the Test2 hashCode() method.

D. class Test2 will not compile.

4.

```
import java.io.*;
public class MyProgram {
    public static void main (String args[]) {
        FileOutputStream out = null;
        try {
            out = new FileOutputStream("test.txt");
            out.write(122);
        } catch (IOException io) {
            System.out.println("IO Error.");
        } finally {
            out.close();
        }
    }
}
```

```

        }
    }
}

```

and given that all methods of class `FileOutputStream`, including `close()`, throw an `IOException`, which of these is true?

- A. This program will compile successfully.
- B. This program fails to compile due to an error at line 4.
- C. This program fails to compile due to an error at line 6.
- D.. This program fails to compile due to an error at line 18.

5. Which statement is true?

- A. A try statement must have at least one corresponding catch block.
- B. Multiple catch statements can catch the same class of exception more than once.
- C. An Error that might be thrown in a method must be declared as thrown by that method, or be handled within that method.
- D.. Except in case of VM shutdown, if a try block starts to execute, a corresponding finally block will always start to execute.

6.

```
public class Test { }
```

What is the prototype of the default constructor?

- A. `Test()`
- B. `Test(void)`
- C.. `public Test()`
- D. `public Test(void)`

7.

Which three are valid method signatures in an interface?

- (1) `private int getArea();`
- (2) `public float getVol(float x);`
- (3) `public void main(String [] args);`
- (4) `public static void main(String [] args);`
- (5) `boolean setFlag(Boolean [] test);`

- A. 1 and 2
- B.. 2, 3 and 5
- C. 3, 4, and 5
- D. 2 and 4

8. What will be the output of the program?

```
class Test {
    static int s;

    public static void main (String [] args) {
        Test p = new Test();
        p.start();
        System.out.println(s);
    }

    void start () {
        int x = 7;
        twice(x);
        System.out.print(x + " ");
    }

    void twice (int x) {
        x = x*2;
        s = x;
    }
}
```

- A. 7 7
- B.. 7 14
- C. 14 0
- D. 14 14

9.

Which method registers a thread in a thread scheduler?

- A. run();
- B. construct();
- C.. start();
- D. register();

10. Which three guarantee that a thread will leave the running state?

- (1) yield()
- (2) wait()
- (3) notify()
- (4) notifyAll()
- (5) sleep(1000)
- (6) aLiveThread.join()
- (7) Thread.killThread()

- A. 1, 2 and 4
- B.. 2, 5 and 6
- C. 3, 4 and 7
- D. 4, 5 and 7

11.

What will be the output of the program?

```
public class BoolTest {
    public static void main (String [] args) {
        int result = 0;

        Boolean b1 = new Boolean("TRUE");
        Boolean b2 = new Boolean("true");
        Boolean b3 = new Boolean("tRuE");
        Boolean b4 = new Boolean("false");

        if (b1 == b2) /* Line 10 */
            result = 1;
        if (b1.equals(b2) ) /* Line 12 */
            result = result + 10;
        if (b2 == b4) /* Line 14 */
            result = result + 100;
        if (b2.equals(b4) ) /* Line 16 */
            result = result + 1000;
        if (b2.equals(b3) ) /* Line 18 */
            result = result + 10000;

        System.out.println( "result = " + result );
    }
}
```

- A. 0

- B. 1
- C. 10
- D.. 10010

12. Which statement is true about a static nested class?

- A. You must have a reference to an instance of the enclosing class in order to instantiate it.
- B.. It does not have access to nonstatic members of the enclosing class.
- C. It's variables and methods must be static.
- D. It must extend the enclosing class.

13. Which constructs an anonymous inner class instance?

- A. `Runnable r = new Runnable() { };`
- B. `Runnable r = new Runnable(public void run() { });`
- C. `Runnable r = new Runnable { public void run(){} };`
- D.. `System.out.println(new Runnable() { public void run() { } });`

14. What will be the output of the program?

```
public class Test {  
    public static void main (String [] args) {  
        int I = 1;  
        do while ( I < 1 )  
            System.out.print( "I is " + I );  
        while ( I > 1 ) ;  
    }  
}
```

- A. I is 1
- B. I is 1 I is 1
- C.. No output is produced.
- D. Compilation error

15. What will be the output of the program?

```
int i = 0;  
while (1) {  
    if (i == 4) {
```

```

        break;
    }
    ++i;
}
System.out.println( "i = " + i );

```

- A. i = 0
- B. i = 3
- C. i = 4
- D.. Compilation fails.

16. What will be the output of the program?

```

public class TestDogs {
    public static void main (String [] args) {
        Dog [][] theDogs = new Dog[3][];
        System.out.println( theDogs[2][0].toString() );
    }
}
class Dog { }

```

- A. null
- B. theDogs
- C. Compilation fails
- D.. An exception is thrown at runtime

17. What will be the output of the program?

```

public class Test {
    public static void main (String args[]) {
        String str = NULL;
        System.out.println( str );
    }
}

```

- A. NULL
- B.. Compile Error
- C. Code runs but no output
- D. Runtime Exception

18. What will be the output of the program?

```

01 import java.util.*;
02 public class NewTreeSet2 extends NewTreeSet {
03     public static void main (String [] args) {
04         NewTreeSet2 t = new NewTreeSet2();
05         t.count();
06     }
07 }
08
09 protected class NewTreeSet {
10     void count () {
11         for (int x = 0; x < 7; x++,x++) {
12             System.out.print( " " + x );
13         }
14     }
15 }

```

- A. 0 2 4
- B. 0 2 4 6
- C. Compilation fails at line 02
- D.. Compilation fails at line 09

19. What will be the output of the program?

```

public abstract class AbstractTest {
    public int getNum() {
        return 45;
    }

    public abstract class Bar {
        public int getNum() {
            return 38;
        }
    }

    public static void main (String [] args) {
        AbstractTest t = new AbstractTest() {
            public int getNum() {
                return 22;
            }
        };
        AbstractTest.Bar f = t.new Bar() {
            public int getNum() {

```



```

        return 57;
    }
};

    System.out.println( f.getNum() + " " + t.getNum() );
}
}

```

- A.. 57 22
- B. 45 38
- C. 45 57
- D. An exception occurs at runtime.

20.

```

01 class HappyGarbage01 {
02     public static void main (String args[]) {
03         HappyGarbage01 h = new HappyGarbage01();
04         h.methodA(); /* Line 04 */
05     }
06     Object methodA() {
07         Object obj1 = new Object();
08         Object [] obj2 = new Object[1];
09         obj2[0] = obj1;
10         obj1 = null;
11         return obj2[0];
12     }
13 }

```

Where will be the most chance of the garbage collector being invoked?

- A. After line 07
- B. After line 08
- C. After line 09
- D.. Garbage collector never invoked in methodA()