SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. You've heard this before, and this time we really mean it this chapter's material is crucial for the exam! Regardless of what the exam question is really testing, there's a good chance that flow control code will be part of the question. Expect to see loops and *if* tests used in questions throughout the entire range of exam objectives.

Flow Control (if and switch) (Sun Objective 2.1)

1. Given the following,

```
1.     public class Switch2 {
2.        final static short x = 2;
3.        public static int y = 0;
4.        public static void main (String [ ] args) {
5.              for (int z = 0; z < 3; z ++) {
6.                 switch (z) {
7.                       case y: System.out.print ("0 ");
8.                       case x-1: System.out.print ("1 ");
9.                       case x:  System.out.print ("2 ");
10.                }
11.             }
12.       }
13.    }
```

What is the result?

A. 0 1 2
B. 0 1 2 1 2 2
C. Compilation fails at line 7
D. Compilation fails at line 8
E. Compilation fails at line 9
F. Ann Exception is thrown at runtime.

2. Given the following,

```
1.     public class Switch2 {
2.        final static short x = 2 ;
3.        public static int y = 0 ;
4.        public static void main (String [ ] args)  {
5.              for (int z = 0; z < 3; z ++) {
6.                 switch (z) {
7.                       case x: System.out.print ("0 ") ;
8.                       case x-1: System.out.print ("1 ") ;
9.                       case x-2: System.out.print ("2 ") ;
10.                }
11.             }
12.       }
13.     }
```

What is the result?

A. 0 1 2
B. 0 1 2 1 2 2
C. 2 1 0 1 0 0
D. 2 1 2 0 1 2
E. Compilation fails at line 8.
F. Compilation fails at line 9.

3. Given the following,

```
1.      public class If1 {
2.          static boolean b;
3.          public static void main (String [ ] args) {
4.                  short hand = 42;
5.                  if ( hand < 50 & !b ) hand ++ ;
6.                  if ( hand > 50 ) ;
7.                  else if ( hand > 40 ) {
8.                      hand += 7 ;
9.                      hand ++ ;       }
10.                  else
11.                      --hand ;
12.                 System.out.println (hand) ;
13.         }
14.     }
```

What is the result?

A. 41
B. 42
C. 50
D. 51
E. Compiler fails at line 5
F. Compiler fails at line 6.

4. Given the following,

```
1.      public class Switch2  {
2.                  final static short x = 2
3.                  public static int y = 0;
4.          public static void main (String [ ] args)  {
5.                  for (int z = 0; z < 4; z++)  {
6.                      switch (z)  {
7.                              case x: System.out.print (“0 “) ;
8.                              default: System.out.print (“def “) ;
9.                              case x – 1: System.out.print (“1 “) ; break;
10.                             case x – 2: System.out.print (“2 “) ;
11.                     }
```

```
            }
        }
    }
```

What is the result?

A. 0 def 1
B. 2 1 0 def 1
C. 2 1 0 def def
D. 2 1 def 0 def 1
E. 2 1 2 0 def 1 2
F. 2 1 0 def 1 def 1

5. Given the following,

```
public class If2  {
    static boolean b1, b2;
    public static void main (String [ ] args)  {
            int x = 0 ;
            if ( !b1 )  {
                if ( !b2 )  {
                        b1 = true ;
                        x++ ;
                        if ( 5 > 6 ) {
                            x++ ;
                        }
                        if ( !b1) x = x + 10;
                        else if ( b2 = true ) x = x + 100 ;
                        else if ( b1 | b2 ) x = x + 1000;
                }
            }
            System.out.println (x) ;
    }
}
```

What is the result?

A. 0
B. 1
C. 101
D. 111
E. 1001
F. 1101

Flow Control (loops) (Sun Objective 2.2)

6. Given the following,

```
public class While  {
    public void loop ( )  {
```

```
3.          int x = 0 ;
4.          while ( 1 )  {
5.             System.out.print (“x plus one is “ + (x + 1) ) ;
6.          }
7.    }
8. }
```

Which statement is true?

A. There is a syntax error on line 1.
B. There are syntax errors on lines 1 and 4.
C. There are syntax errors on lines 1, 4 and 5.
D. There is a syntax error on line 4.
E. There are syntax errors on lines 4 and 5.
F. There is a syntax error on line 5.

7. Given the following,

```
1. class For  {
2.    public void test ( )  {
3.
4.             System.out.println (“x = “+ x);
5.          }
6.    }
7. }
```

And the following output,

```
x = 0
x = 1
```

which two lines of code (inserted independently) will cause this output? (Choose two.)

A. for (int x = -1; x < 2; ++x) {
B. for (int x = 1; x < 3; ++x) {
C. for (int x = 0; x > 2; ++x) {
D. for (int x = 0; x < 2; x++) {
E. for (int x = 0; x < 2; ++x) {

8. Given the following,

```
1. public class Test  {
2.    public static void main (String [ ] args)  {
3.          int I = 1;
4.          do while ( I < 1)
5.             System.out.print (“I is “ + I) ;
6.          while ( I > 1 ) ;
7.    }
8. }
```

What is the result?

A. I is 1
B. I is 1 I is 1
C. No output is produced
D. Compilation error
E. I is 1 I is 1 I is 1 in an infinite loop.

9. Given the following,

```
int I = 0;
outer:
    while (true)  {
        I++ ;
        inner:
            for (int j = 0; j < 10; j ++)  {
            I += j;
            if (j = = 3)
                continue inner;
            break outer;
        }
        continue outer;
    }
System.out.println (I) ;
```

What is the result?

A. 1
B. 2
C. 3
D. 4

10. Given the following,

```
int I = 0;
label:
    if (I < 2)  {
        Syste,.out.print (“I is “ + I) ;
        I++;
        continue label;
    }
```

What is the result?

A. I is 0
B. I is 0 I is 1
C. Compilation fails.

D. None of the above

Exceptions (Sun Objectives 2.3 and 2.4)

11. Given the following,

```
System.out.print (“Start “ ) ;
try  {
        System.out.print (“Hello world”) ;
        throw new FileNotFoundException ( ) ;
}
System.out.print (“ Catch Here “ ) ;
catch (EOFException e)  {
        System.out.print (“End of file exception”) ;
}
catch (FileNotFoundException e)  {
        System.out.print (“File not found”) ;
}
```

And given that EOFException and FileNotFoundException are both subclasses of IOException, and further assuming this block of code is placed into a class, which statement is most true concerning this code?

A. The code will not compile.
B. Code output: Start Hello world File Not Found.
C. Code output: Start Hello world End of file exception.
D. Code output: Start Hello world Catch Here File not found.

12. Given the following,

```
public class MyProgram  {
   public static void main (String args [ ] )  {
        try  {
           System.out.print (“Hello world “) ;
        }
        finally  {
           System.out.println (“Finally executing “ ) ;
        }
   }
}
```

What is the result?

A. Nothing. The program will not compile because no exceptions are specified.
B. Nothing. The program will not compile because no catch clauses are specified.
C. Hello World
D. Hello world Finally executing

13. Given the following,

```
1.     import java.io.*;
2.     public class MyProgram  {
3.        public static void main (String args [ ] )  {
4.              FileOutputStream out = null ;
5.              try  {
6.                 out = new FileOutputStream (“test.txt”) ;
7.                 out.write (122);
8.              }
9.              catch (IOException io)  {
10.                System.out.println (“ IO Error.”) ;
11.             }
12.             finally  {
13.                out.close ( ) ;
14.             }
15.       }
16.    }
```

And given that all methods of class FileOutputStream, including close ( ), throw an IOException, which of these is true? (Choose one.)

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 9
E. This program fails to compile due to an error at line 13.

14. Given the following,

```
1.     public class MyProgram  {
2.        public static void throwit ( )  {
3.              throw new RuntimeException ( ) ;
4.        }
5.        public static void main (String args [ ] )  {
6.              try  {
7.                 System.out.println (“Hello world “) ;
8.                 throwit ( ) ;
9.                 System.out.println (“Done with try block “) ;
10.             }
11.             finally  {
12.                System.out.println (“Finally executing “) ;
13.             }
14.       }
15.    }
```

Which answer most closely indicates the behavior of the program?

A. The program will not compile.

B. The program will print Hello world, then will print that a Runtime Exception has occurred, then will print Done with try block, and then will print Finally executing.

C. The program will print Hello world, then will print that a RuntimeException has occurred, and then will print Finally executing.

D. The program will print Hello world, then will print Finally executing, then will print that a RuntimeException has occurred.

15. Given the following,

```
public class RTExcept  {
   public static void throwit ( )  {
         System.out.print (“throwit “);
         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 “) ;
   }
}
```

What is the result?

A. hello throwit caught
B. Compilation fails
C. hello throwit RuntimeException caught after
D. hello throwit RuntimeException
E. hello throwit caught finally after
F. hello throwit caught finally after RuntimeException

Assertions (Sun Objectives 2.5.and 2.6)

16. Which of the following statements is true?

A. In an *assert* statement, the expression after the colon (:) can be *any* Java expression.

B. If a *switch* block has no default, adding an *assert* default is considered appropriate.

C. In an *assert* statement, if the expression after the colon (:) does not have a value, the assert’s error message will be empty.

D. It is appropriate to handle assertion failures using a catch clause.

17. Which two of the following statements are true? (Choose Two.)

A. It is sometimes good practice to throw an AssertionError explicitly
B. It is good practice to place assertions where you think execution should never reach.
C. Private getter ( ) and setter ( ) methods should not use assertions to verify arguments.
D. If an AssertionError is thrown in a try – catch block, the finally block will be bypassed.
E. It is proper to handle assertion statement failures using a *catch* (AssertionException ae) block.

18. Given the following,

```
public class Test  {
   public static int y;
   public static void foo (int x)  {
         System.out.print (“foo “) ;
         y = x ;
   }
   public static int bar (int z)  {
         System. out.print (“bar “) ;
         return y = z ;
   }
   public static void main (String [ ] args )  {
         int t = 0;
         assert t > 0 : bar (7);
         assert t > 1 : foo (8);
         System.out.println (“done “) ;
   }
}
```

What is the result?

A. bar
B. bar done
C. foo done
D. bar foo done
E. Compilation fails
F. An error is thrown at runtime.

19. Which two of the following statements are true? (Choose two.)

A. If assertions are compiled into a source file, and if no flags are included at runtime, assertions will execute by default.
B. As of Java version 1.4, assertion statements are compiled by default.
C. With the proper use of runtime arguments, it is possible to instructor the VM to disable assertions for a certain class, and to enable assertions for a certain package, at the same time.

D. The following are all valid runtime assertion flags:
-ea, -esa, -dsa, -enableassertions,
-disablesystemassertions

E. When evaluating command-line arguments, the VM gives –ea flags precedence over –da flags.

20. Given the following,

```
1.      public class Test2  {
2.          public static int x;
3.          public static int foo (int y)  {
4.                  return y * 2;
5.          }
6.          public static void main (String [ ] args)  {
7.                  int z = 5;
8.                  assert z > 0 ;
9.                  assert z > 2 : foo (z) ;
10.                 if ( z < 7 )
11.                 assert z > 4 ;
12.                 switch (z)  {
13.                     case 4:  System.out.println (“4 “) ;
14.                     case 5:  System.out.println (“5 “) ;
15.                     default: assert z < 10 ;
16.                 }
17.                 if ( z < 10 )
18.                     assert z > 4: z ++;
19.                 System.out.println (z);
20.         }
21.     }
```

Which line is an example of an inappropriate use of assertions?

A. Line 8
B. Line 9
C. Line 11
D. Line 15
E. Line 18