MULTIPLE-CHOICE QUESTIONS ON INTRODUCTORY JAVA LANGUAGE CONCEPTS

1. Which of the following pairs of declarations will cause an error message?

```
I   double x = 14.7;
    int y = x;

II  double x = 14.7;
    int y = (int) x;

III int x = 14;
    double y = x;
```

(A) None
(B) I only
(C) II only
(D) III only
(E) I and III only

2. What output will be produced by

```
System.out.print("\\* This is not\n a comment *\\");
```

(A) `* This is not a comment *`

(B) `\* This is not a comment *\`

(C)
```
* This is not
a comment *
```

(D)
```
\\* This is not
a comment *\\
```

(E)
```
\* This is not
a comment *\
```

3. Consider the following code segment

```
if (n != 0 && x / n > 100)
    statement1;
else
    statement2;
```

If `n` is of type `int` and has a value of `0` when the segment is executed, what will happen?

(A) An `ArithmeticException` will be thrown.
(B) A syntax error will occur.
(C) ***statement1***, but not ***statement2***, will be executed.
(D) ***statement2***, but not ***statement1***, will be executed.
(E) Neither ***statement1*** nor ***statement2*** will be executed; control will pass to the first statement following the `if` statement.

8. Let x be a variable of type double that is positive. A program contains the boolean expression (Math.pow(x,0.5) == Math.sqrt(x)). Even though $x^{1/2}$ is mathematically equivalent to $\sqrt{x}$, the above expression returns the value false in a student's program. Which of the following is the most likely reason?

(A) Math.pow returns an int, while Math.sqrt returns a double.
(B) x was imprecisely calculated in a previous program statement.
(C) The computer stores floating-point numbers with 32-bit words.
(D) There is round-off error in calculating the pow and sqrt functions.
(E) There is overflow error in calculating the pow function.

9. What will the output be for the following poorly formatted program segment, if the input value for num is 22?

```
int num =  call to a method that reads an integer;
if (num > 0)
if (num % 5 == 0)
System.out.println(num);
else System.out.println(num + " is negative");
```

(A) 22
(B) 4
(C) 2 is negative
(D) 22 is negative
(E) Nothing will be output.

10. What values are stored in x and y after execution of the following program segment?

```
int x = 30, y = 40;
if (x >= 0)
{
    if (x <= 100)
    {
        y = x * 3;
        if (y < 50)
            x /= 10;
    }
    else
        y = x * 2;
}
else
    y = -x;
```

(A) x = 30 y = 90
(B) x = 30 y = -30
(C) x = 30 y = 60
(D) x = 3 y = -3
(E) x = 30 y = 40

15. Given that `n` and `count` are both of type `int`, which statement is true about the following code segments?

```
I for (count = 1; count <= n; count++)
      System.out.println(count);

II count = 1;
   while (count <= n)
   {
       System.out.println(count);
       count++;
   }
```

(A) I and II are exactly equivalent for all input values `n`.
(B) I and II are exactly equivalent for all input values $n \geq 1$, but differ when $n \leq 0$.
(C) I and II are exactly equivalent only when $n = 0$.
(D) I and II are exactly equivalent only when `n` is even.
(E) I and II are not equivalent for any input values of `n`.

16. The following fragment intends that a user will enter a list of positive integers at the keyboard and terminate the list with a sentinel:

```
int value = 0;
final int SENTINEL = -999;
while (value != SENTINEL)
{
    //code to process value
      ...
    value = IO.readInt();       //read user input
}
```

The fragment is not correct. Which is a true statement?
(A) The sentinel gets processed.
(B) The last nonsentinel value entered in the list fails to get processed.
(C) A poor choice of `SENTINEL` value causes the loop to terminate before all values have been processed.
(D) The code will always process a value that is not on the list.
(E) Entering the `SENTINEL` value as the first value causes a run-time error.

17. Suppose that base-2 (binary) numbers and base-16 (hexadecimal) numbers can be denoted with subscripts, as shown below:

$$2A_{\text{hex}} = 101010_{\text{bin}}$$

Which is equal to $3D_{\text{hex}}$?
(A) 111101_{bin}
(B) 101111_{bin}
(C) 10011_{bin}
(D) 110100_{bin}
(E) 101101_{bin}

Questions 21 and 22 refer to the following method, `checkNumber`, which checks the validity of its four-digit integer parameter.

```
/** @param n a 4-digit integer
 *  @return true if n is valid, false otherwise
 */
boolean checkNumber(int n)
{
    int d1,d2,d3,checkDigit,nRemaining,rem;
    //strip off digits
    checkDigit = n % 10;
    nRemaining = n / 10;
    d3 = nRemaining % 10;
    nRemaining /= 10;
    d2 = nRemaining % 10;
    nRemaining /= 10;
    d1 = nRemaining % 10;
    //check validity
    rem = (d1 + d2 + d3) % 7;
    return rem == checkDigit;
}
```

A program invokes method `checkNumber` with the statement

```
boolean valid = checkNumber(num);
```

21. Which of the following values of `num` will result in `valid` having a value of `true`?
    - (A) `6143`
    - (B) `6144`
    - (C) `6145`
    - (D) `6146`
    - (E) `6147`

22. What is the purpose of the local variable `nRemaining`?
    - (A) It is not possible to separate `n` into digits without the help of a temporary variable.
    - (B) `nRemaining` prevents the parameter `num` from being altered.
    - (C) `nRemaining` enhances the readability of the algorithm.
    - (D) On exiting the method, the value of `nRemaining` may be reused.
    - (E) `nRemaining` is needed as the left-hand side operand for integer division.

24. Which of the following program fragments will produce this output? (Ignore spacing.)

```
2 - - - - -
- 4 - - - -
- - 6 - - -
- - - 8 - -
- - - - 10 -
- - - - - 12
```

```
I for (int i = 1; i <= 6; i++)
  {
      for (int k = 1; k <= 6; k++)
          if (k == i)
              System.out.print(2 * k);
          else
              System.out.print("-");
      System.out.println();
  }

II for (int i = 1; i <= 6; i++)
   {
       for (int k = 1; k <= i - 1; k++)
           System.out.print("-");
       System.out.print(2 * i);
       for (int k = 1; k <= 6 - i; k++)
           System.out.print("-");
       System.out.println();
   }

III for (int i = 1; i <= 6; i++)
    {
        for (int k = 1; k <= i - 1; k++)
            System.out.print("-");
        System.out.print(2 * i);
        for (int k = i + 1; k <= 6; k++)
            System.out.print("-");
        System.out.println();
    }
```

(A) I only
(B) II only
(C) III only
(D) I and II only
(E) I, II, and III

26. Consider the method reverse:

```
/** Precondition:  n > 0.
 *  Postcondition:
 *   - Returns n with its digits reversed.
 *   - Example: If n = 234, method reverse returns 432.
 *  @param n a positive integer
 *  @return n with its digits reversed
 */
int reverse(int n)
{
    int rem, revNum = 0;

    /* code segment */

    return revNum;
}
```

Which of the following replacements for `/* code segment */` would cause the method to work as intended?

```
I   for (int i = 0; i <= n; i++)
    {
        rem = n % 10;
        revNum = revNum * 10 + rem;
        n /= 10;
    }

II  while (n != 0)
    {
        rem = n % 10;
        revNum = revNum * 10 + rem;
        n /= 10;
    }

III for (int i = n; i != 0; i /= 10)
    {
        rem = i % 10;
        revNum = revNum * 10 + rem;
    }
```

(A) I only
(B) II only
(C) I and II only
(D) II and III only
(E) I and III only