Practice Exam #2

SECTION I

Time — 1 hour and 15 minutes
Number of questions — 40
Percent of total grade — 50

1. What is the output of the following program segment?

```
int num = 5;
while (num >= 0)
{
  num -= 2;
}
System.out.print(num);
```

(A) -2
(B) -1
(C) 0
(D) 2
(E) 21

2. What is the output from

```
int n = 12;
System.out.print(goFigure(n));
System.out.print(" " + n);
```

where the method `goFigure` is defined as follows:

```
public double goFigure(int n)
{
  n %= 7;
  return (double)(12 / n);
}
```

(A) 2.0 12
(B) 2.4 12
(C) 2.0 5
(D) 2.4 5
(E) 2.4 6

3. Which of the following expressions will evaluate to `true` when `x` and `y` are `boolean` variables with different values?

I. `(x || y) && (!x || !y)`
II. `(x || y) && !(x && y)`
III. `(x && !y) || (!x && y)`

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

4. What is the output from the following code segment?

```
double x = 5*4/2 - 5/2*4;
System.out.println(x);
```

(A) `0`
(B) `1`
(C) `0.0`
(D) `1.0`
(E) `2.0`

5. What is the output from the following code segment?

```
String band = "anamanaguchi";
System.out.println(band.substring(1, 4).
                        compareTo(band.substring(5, 8)));
```

(A) `true`
(B) `false`
(C) `0`
(D) A negative integer
(E) A positive integer

6. Integers can be represented in number systems with different bases. For example, $34_{dec} = 22_{hex} = 100010_{bin}$, where the subscripts "dec," "hex," and "bin" indicate the decimal (base 10), hexadecimal (base 16), and binary (base 2) systems, respectively. What are the hexadecimal and binary representations of 49_{dec}?

(A) $49_{dec} = 21_{hex} = 110011_{bin}$
(B) $49_{dec} = 31_{hex} = 11001_{bin}$
(C) $49_{dec} = 31_{hex} = 110001_{bin}$
(D) $49_{dec} = 37_{hex} = 110111_{bin}$
(E) $49_{dec} = 61_{hex} = 110001_{bin}$

Questions 7-8 refer to the following method.

```
public void printVals(String[] items, int k)
{
  if (k > 1)
  {
    printVals(items, k - 1);
    System.out.print(items[k] + " ");
    printVals(items, k - 2);
  }
}
```

Suppose the following code segment has been executed.

```
String[] names = {"Pat", "Joe", "Ann", "Cal", "Amy"};
printVals(names, names.length - 1);
```

7. What is the output?

(A) `Ann Cal Amy Ann`
(B) `Ann Cal Amy Cal Ann`
(C) `Ann Cal Joe Amy Joe Ann`
(D) `Joe Ann Cal Joe Amy Joe Ann`
(E) `Joe Ann Pat Cal Joe Amy Joe Ann Pat`

8. How many calls to `printVals` have been made, including the original call?

(A) 3
(B) 5
(C) 7
(D) 8
(E) 9

9. Consider the following code segment.

```
String abc = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
< missing statement >
System.out.println(abc.substring(k, k+1));
```

It is supposed to print a randomly chosen letter of the alphabet. Any of the 26 letters can be chosen with equal probability. Which of the following can replace *< missing statement >* for the code to work that way?

(A) `int k = Math.random(25);`
(B) `int k = Math.random(0, 25);`
(C) `int k = Math.random(25) + 1;`
(D) `int k = Math.random(26) - 1;`
(E) `int k = (int)(26*Math.random());`

10. In OOP, programmers often arrange classes into inheritance hierarchies instead of implementing isolated classes. Which of the following is NOT a valid reason for doing so?

    (A) Abstract classes at the top of the hierarchy can be extended in the project or reused in other projects.
    (B) Methods from a superclass can often be reused in its subclasses without duplication of code.
    (C) Objects from different subclasses can be passed as parameters to a method designed to accept objects of the superclass.
    (D) Objects from different subclasses can be stored in the same array.
    (E) All of the above are valid reasons for using inheritance hierarchies.

11. At the county fair, prizes are awarded to the five heaviest cows. More than 2000 cows are entered, and their records are stored in an array. Which of the following algorithms provides the most efficient way of finding the records of the five heaviest cows?

    (A) Selection Sort
    (B) Selection Sort terminated after the first five iterations
    (C) Insertion Sort
    (D) Insertion Sort terminated after the first five iterations
    (E) Mergesort

12. Which of the following recommendations for testing software is NOT good advice?

    (A) Test a program with all possible values of input data.
    (B) When testing a large program, test the smaller pieces individually before testing the entire program.
    (C) If possible, use automated testing procedures or read test data from files so that you can re-run the tests after corrections have been made.
    (D) Design test data that exercises as many different paths through the code as is practical.
    (E) Test on data that is at the boundary of program conditionals to check for "off by one" errors.

13. Given

```
String s = "ALASKA";
int index = s.substring(1, 4).indexOf("A");
```

    what is the value of `index`?

    (A) 0
    (B) 1
    (C) 2
    (D) 5
    (E) -1

14. Consider the following method of the class `Test`.

```
public static List<String> doNothing(List<String> list)
{
  return list;
}
```

Which of the following program segments in a `Test`'s client class will compile with no errors?

I.
```
ArrayList<String> nums = new ArrayList<String>();
nums = Test.doNothing(nums);
```

II.
```
List<String> nums = new ArrayList<String>();
nums = Test.doNothing(nums);
```

III.
```
ArrayList<String> nums1 = new ArrayList<String>();
List<String> nums2 = Test.doNothing(nums1);
```

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

15. What will array `arr` contain after the following code segment has been executed?

```
int[] arr = {4, 3, 2, 1, 0};
for (int k = 1; k < arr.length; k++)
{
  arr[k-1] += arr[k];
}
```

(A) 4, 7, 5, 3, 1
(B) 4, 7, 9, 10, 10
(C) 7, 5, 3, 1, 0
(D) 7, 3, 2, 1, 0
(E) 10, 6, 3, 1, 0

Questions 16-17 refer to the following method.

```
private int product(int n)
{
  if (n <= 1)
    return 1;
  else
    return n * product(n-2);
}
```

16. What is the output when `product(6)` is called?

(A) 1
(B) 8
(C) 12
(D) 48
(E) 720

17. `product(25)` returns −1181211311, a negative number. Which of the following accounts for this result?

(A) Integer arithmetic overflow
(B) Logic error that shows up for odd values of `n`
(C) Stack overflow error in recursive calls
(D) Small range of integers in the Java Virtual Machine installed on your computer
(E) A loss of precision in calculations

18. An "assertion" in a program is

(A) a comment in the source code of the program that documents that a certain condition is satisfied when the program reaches a particular point.
(B) a directive to the compiler to verify that a particular variable is initialized to a specified value at a given point in the source code.
(C) an executable statement within a loop that checks whether a specified condition is satisfied; if not, the loop is broken and control is passed to the first statement after the loop.
(D) an executable statement in the program that displays a message if a method's precondition or postcondition is not satisfied.
(E) an executable statement in the program that aborts the program if a specified condition is not satisfied.

19. The two versions of the `search` method shown below are both intended to return `true` if `ArrayList<Object> list` contains the target value; otherwise they are supposed to return `false`.

Version 1

```
public boolean search(ArrayList<Object> list, Object target)
{
  for (Object x : list)
  {
    if (target.equals(x))
      return true;
  }
  return false;
}
```

Version 2

```
public boolean search(ArrayList<Object> list, Object target)
{
  boolean found = false;

  for (Object x : list)
  {
    if (target.equals(x))
      found = true;
  }
  return found;
}
```

Which of the following statements about the two versions of `search` is true?

(A) Only Version 1 works as intended.
(B) Only Version 2 works as intended.
(C) Both versions work as intended; Version 1 is often more efficient than Version 2.
(D) Both versions work as intended; Version 2 is often more efficient than Version 1.
(E) Both versions work as intended; the two versions are always equally efficient.

20. Consider the following class.

```
public class BuddyList
{
  /** Contains the names of buddies */
  private ArrayList<String> buddies;

  < constructors and other methods and variables not shown >

  public ArrayList<String> getBuddies()
  {
    return buddies;
  }
}
```

If `BuddyList myFriends` is declared and initialized in some other class, a client of `BuddyList`, which of the following correctly assigns to `name` the name of the first buddy in the `myFriends` list?

I. `String name = myFriends.buddies[0];`

II. `String name = myFriends.buddies.get(0);`

III. `String name = myFriends.getBuddies().get(0);`

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

21. The code fragment

```
int x = < an integer value > ;
System.out.println(x*x);
```

displays `-131071`. Which of the following is a possible value of `x`?

(A) -1
(B) $2^{15} + 1$
(C) $2^{16} - 1$
(D) $2^{32} + 1$
(E) No such value exists

22. What is the value of `product` after the following code segment is executed?

```
int[] factors = {2, 3, 4, 7, 2, 5};
int product = 1;

for (int i = 1; i < factors.length; i += 2)
{
  product *= (factors[i] % factors[i - 1]);
}
```

(A) 0
(B) 1
(C) 2
(D) 3
(E) 5

23. What is the output of the following code segment?

```
List<Integer> list = new ArrayList<Integer>();

for (int i = 1; i <= 8; i++)
{
  list.add(new Integer(i));
}

for (int i = 0; i < list.size(); i++)
{
  list.remove(i);
}

for (Integer x : list)
{
  System.out.print(x + " ");
}
```

(A) `IndexOutOfBoundsException`
(B) No output because the resulting list is empty
(C) `1 3 5 7`
(D) `2 4 6 8`
(E) `1 2 3 4 5 6 7 8`

24. Consider the following method.

```
public void change(double[] nums, int n)
{
  for (int k = 0; k < n; k++)
  {
    nums[k] = 5.4;
  }
  n = 2;
}
```

What will be stored in `samples` and `len` after the following statements are executed?

```
double[] samples = {1.0, 2.1, 3.2, 4.3};
int len = samples.length;
change(samples, len);
```

(A) `samples` contains `5.4, 5.4, 5.4, 5.4`; `len` is `4`
(B) `samples` contains `5.4, 5.4, 5.4, 5.4`; `len` is `2`
(C) `samples` contains `1.0, 2.1, 3.2, 4.3`; `len` is `4`
(D) `samples` contains `5.4, 5.4`; `len` is `2`
(E) `samples` contains `1.0, 2.1`; `len` is `2`

25. Consider the following code segment.

```
List<String> list = new ArrayList<String>();
list.add("A");
list.add("B");
list.add("C");
for (String s : list)
{
  String t = list.get(list.size() - 1);
  list.set(list.size() - 1, s);
  s = t;
}
```

What will `list` contain after the above code segment has been executed?

(A) `["A", "B", "C"]`
(B) `["A", "B", "B"]`
(C) `["C", "B", "A"]`
(D) `["C", "A", "B"]`
(E) `["C", "C", "C"]`

26. Consider the following class.

```
public class Counter
{
  private int count = 0;

  public Counter() { count = 0; }
  public Counter(int x) { count = x; }              // Line 1
  public int getCount() { return count; }           // Line 2
  public void setCount(int c) { int count = c; }    // Line 3
  public void increment() { count++; }              // Line 4
  public String toString() { return "" + count; }   // Line 5
}
```

The test code

```
Counter c = new Counter();
c.setCount(3);
c.increment();
System.out.println(c.getCount());
```

is supposed to print `4`, but the class has an error. What is actually printed, and which line in the class definition should be changed to get the correct output, `4`?

(A) 0, Line 1
(B) 1, Line 3
(C) 0, Line 4
(D) 3, Line 4
(E) 36, Line 5

27. Suppose an array `arr` contains 127 different random values arranged in ascending order, and the most efficient searching algorithm is used to find a target value. How many elements of the array will be examined when the target equals `arr[39]`?

(A) 4
(B) 5
(C) 7
(D) 63
(E) 64

28. Consider the following interface and class.

```
public interface Student
{
  double getGPA();
  int getSemesterUnits();
}

public class FullTimeStudent implements Student
{
  < required methods go here >
}
```

What is the minimum set of methods that a developer must implement in order to successfully compile the `FullTimeStudent` class?

(A) No methods would need to be implemented
(B) `getGPA(), getSemesterUnits()`
(C) `getGPA(), getSemesterUnits(), equals(Object s)`
(D) `getGPA(), getSemesterUnits(), equals(Object s), toString()`
(E) `getGPA(), getSemesterUnits(), compareTo(Object s), equals(Object s), toString()`

29. Consider the following method.

```
public int locate(String str, String oneLetter)
{
  int j = 0;
  while (j < str.length() &&
         str.substring(j, j+1).compareTo(oneLetter) < 0)
  {
    j++;
  }
  return j;
}
```

Which of the following must be true when the `while` loop terminates?

(A) `j == str.length()`
(B) `str.substring(j, j+1) >= 0`
(C) `j <= str.length() || str.substring(j, j+1).compareTo(oneLetter) > 0`
(D) `j == str.length() || str.substring(j, j+1).compareTo(oneLetter) >= 0`
(E) `j == str.length() && str.substring(j, j+1).compareTo(oneLetter) >= 0`

30. Which of the following code segments correctly traverses row by row a rectangular two-dimensional `int` array `m`?

(A)

```
for (int x : m)
{
  ...
}
```

(B)

```
for (int r : m)
{
  for (int c = 0; c < m.length; c++)
  {
    int x = m[r][c];
    ...
  }
}
```

(C)

```
for (int c = 0; c < m.length; c++)
{
  for (int r = 0; r < m[c].length; r++)
  {
    int x = m[r][c];
    ...
  }
}
```

(D)

```
for (int[] r : m)
{
  for (int c = 0; c < m[0].length; c++)
  {
    int x = r[c];
    ...
  }
}
```

(E)

```
for (int r = 0; r < m[0].length; r++)
{
  for (int c = 0; c < m.length; c++)
  {
    int x = m[r][c];
    ...
  }
}
```

31. The following incomplete method `cutToAverage` first finds the average `avg` of the values in an array, then replaces every element that exceeds `avg` with `avg`. `cutToAverage` returns `avg`.

```
public double cutToAverage(double[] amps)
{
  double avg = 0.0;
  < missing code >
  return avg;
}
```

Which of the following could replace < *missing code* > so that `cutToAverage` works as specified?

I.
```
for (double x : amps)
  avg += x;
avg /= amps.length;
for (double x : amps)
  if (x > avg)
    x = avg;
```

II.
```
for (int k = 0; k < amps.length; k++)
  avg += amps[k];
avg /= amps.length;
for (int k = 0; k < amps.length; k++)
  if (amps[k] > avg)
    amps[k] = avg;
```

III.
```
for (double x : amps)
  avg += x;
avg /= amps.length;
for (int k = amps.length - 1; k >= 0; k--)
  if (amps[k] > avg)
    amps[k] = avg;
```

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

32. Consider the following class `Athlete`.

```
public class Athlete implements Comparable<Athlete>
{
  private int numMedals;

  public int getRank() { return numMedals; }

  public int compareTo(Athlete other)
  {
    // return numMedals - other.numMedals;
    return getRank() - other.getRank();
  }

  < constructors and other methods not shown >
}
```

As you can see, the programmer has commented out direct references to `Athlete`'s instance variable `numMedals` in the `compareTo` method and replaced them with calls to the `getRank` method. What is the most compelling reason for doing this?

(A) To correct a syntax error: being private, neither `numMedals` nor `other.numMedals` are directly accessible in the method's code
(B) To correct a syntax error: being private, `other.numMedals` is not directly accessible in the method's code (`numMedals` is replaced with `getRank()` for consistency)
(C) To avoid possible problems later: if `other` happens to be an object of a subclass of `Athlete` in which `numMedals` is not used in calculating the rank, the original code would fail
(D) To improve run-time efficiency
(E) To achieve better encapsulation of the `Athlete` class

33. What is the result when the following statement is compiled/executed?

```
System.out.println(Math.pow(2, 3));
```

(A) `8` is displayed
(B) `8.0` is displayed
(C) `9` is displayed
(D) Syntax error: "pow(double, double) in java.lang.Math cannot be applied to (int, int)"
(E) `ArithmeticException` at run time

Questions 34-37 refer to the following classes.

```
public class Party
{
  private List<String> theGuests;

  public Party() { theGuests = null; }

  public Party(List<String> guests) { theGuests = guests; }

  public void setGuests(List<String> guests)
  { theGuests = guests; }

  public String toString()
  { /* implementation not shown */ }
}

public class BDayParty extends Party
{
  private String theName;

  public BDayParty(String name, List<String> guests)
  { /* implementation not shown */ }

  public String getName() { return theName; }

  < other methods not shown >
}
```

34. Given

```
List<String> guests = new ArrayList<String>();
guests.add("Alice");
guests.add("Ben");
guests.add("Candy");
```

which of the following declarations is NOT valid?

(A)
```
Party[] celebrations = new Party[2];
```

(B)
```
Party[] celebrations =
  {new Party(guests), new Party()};
```

(C)
```
BDayParty[] celebrations =
  {new BDayParty("Malika", guests), new Party(guests)};
```

(D)
```
BDayParty[] celebrations =
  {new BDayParty("Lee", guests),
   new BDayParty("Henry", guests)};
```

(E) All of the above are valid.

35. Which of the following statements can replace < *missing statement* > in the following `BDayParty` constructor?

```
public BDayParty(String name, List guests)
{
  < missing statement >
  theName = name;
}
```

I. `theGuests = guests;`

II. `super(guests);`

III. `setGuests(guests);`

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

36. Suppose we have decided to make the `Party` class `abstract` and have added the following methods to it:

```
public abstract String getOccasion();
public String getMessage() { return "Happy"; }
public String greetings() { return getMessage() + " "
                                    + getOccasion(); }
```

Which of the following is the smallest set of `Party` methods that would have to be overridden in the `BDayParty` class to make

```
BDayParty birthday = new BDayParty("Aaron", guests);
System.out.println(birthday.greetings());
```

display

```
Happy Birthday Aaron
```

(A) None
(B) `getOccasion`
(C) `getMessage`
(D) `getOccasion` and `getMessage`
(E) `getOccasion`, `getMessage`, and `greetings`

37. `Party`'s `toString` method lists all the entries in the `theGuests` list. Should the programmer use a "for each" loop or the list's `get(i)` method within a loop to traverse the list?

(A) `get(i)`, because it is always more efficient
(B) `get(i)`, because "for each" loops work only for arrays
(C) A "for each" loop, because `get(i)` may be not available in the implementation of `List` passed to the constructor
(D) A "for each" loop, because it is more efficient when `theGuests` happens to be an `ArrayList`
(E) Either method works and is equally efficient when the `List` passed to the constructor is an `ArrayList`.

38. Suppose `mat` is declared as

```
int[][] mat = new int[3][4];
```

If `mat` initially contains

```
2 1 3 4
9 7 2 1
0 2 5 6
```

what is the output of the following code segment?

```
for (int r = 1; r < mat.length; r++)
{
  for (int c = 1; c < mat[0].length; c++)
  {
    if ((r + c) % 2 == 0)
      mat[r][c] = 2 * mat[r - 1][c - 1] + c;
  }
}
System.out.println(mat[2][2]);
```

(A) 5
(B) 11
(C) 12
(D) 15
(E) 16

39. Suppose it takes about 18 milliseconds to sort an array of 80,000 random numbers using Mergesort. Suppose for an array of 160,000 numbers, Mergesort runs for 40 milliseconds. Approximately how much time will Mergesort run on an array of 320,000 numbers? Choose the closest estimate.

    (A) 80 milliseconds
    (B) 88 milliseconds
    (C) 96 milliseconds
    (D) 126 milliseconds
    (E) 160 milliseconds

40. Suppose a class `Particle` has the following variables defined.

```
public class Particle
{
  public static final int START_POS = 100;
  private double velocity;

  private boolean canMove()
  { /* implementation not shown */ }

  < constructors, other variables and methods not shown >
}
```

    Which of the following is true?

    (A) Java syntax rules wouldn't allow us to use the name `startPos` instead of `START_POS`.

    (B) Java syntax rules wouldn't allow us to make `velocity` public.

    (C) Both `velocity` and `START_POS` can be changed by one of `Particle`'s methods.

    (D) A statement

```
double pos = START_POS + velocity;
```

    in `Particle`'s `canMove` method would result in a syntax error.

    (E) All of the above statements are false.

Answers and Solutions

Exam #2 ~ Multiple Choice

1. B	11. B	21. C	31. D
2. A	12. A	22. D	32. C
3. E	13. B	23. D	33. B
4. E	14. D	24. A	34. C
5. E	15. C	25. B	35. D
6. C	16. D	26. B	36. B
7. A	17. A	27. A	37. E
8. E	18. E	28. B	38. C
9. E	19. C	29. D	39. B
10. E	20. C	30. D	40. E

Notes:

1. The last iteration starts with `num` = 1.
2. `12%7` gives 5; `(double)(12/5)` results in 2.0; `n` remains unchanged because it is passed to `goFigure` by value.
3. Applying one of De Morgan's Laws, we can see that the expressions in Options I and II are identical. The expression in Option III is `true` when `x ≠ y`, the same as I and II.
4. `5*4/2 = (5*4)/2` evaluates to 10. `5/2*4 = (5/2)*4` evaluates to 8. The result is converetd into `double` at the end.
5. `band.substring(1, 4)` is `"nam"`; `band.substring(5, 8)` is `"nag"`.
6. 49 = 3*16 + 1; Hex didit 3 is 0011 in binary; hex digit 1 is 0001 in binary.
7. `printVals(names, 0)` and `printVals(names, 1)` do nothing.

   `printVals(names, 2)` prints `"Ann"`.

   `printVals(names, 3)` prints `"Ann"`, then `"Cal"`.

   `printVals(names, 4)` prints `"Ann"`, then `"Cal"`, then `"Amy"`, then `"Ann"`.
8. `printVals(names, 2)` results in 1 + 1 + 1 = 3 calls.

   `printVals(names, 3)` results in 1 + 3 + 1 = 5 calls.

   `printVals(names, 4)` results in 1 + 5 + 3 = 9 calls.
9. `Math.random` takes no parameters and returns a random `double` in the range $0 \le x < 1$.
10. Inheritance hierarchies are at the heart of OOP for the reasons listed in the question.
11. Selection Sort finds the largest element among 2000, then among the remaining 1999, and so on.