Practice Exam One

COMPUTER SCIENCE

SECTION I

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

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. Do not spend too much time on any one problem.

Notes:

• Assume that the classes in the Quick Reference have been imported where needed.
• Assume that variables and methods are declared within the context of an enclosing class.
• Assume that method calls that have no object or class name prefixed, and that are not shown within a complete class definition, appear within the context of an enclosing class.
• Assume that parameters in method calls are not null unless otherwise stated.
1. Consider this inheritance hierarchy, in which Novel and Textbook are subclasses of Book.

Which of the following is a false statement about the classes shown?
(A) The Textbook class can have private instance variables that are in neither Book nor Novel.
(B) Each of the classes—Book, Novel, and Textbook—can have a method computeShelfLife, whose code in Book and Novel is identical, but different from the code in Textbook.
(C) If the Book class has private instance variables myTitle and myAuthor, then Novel and Textbook inherit them but cannot directly access them.
(D) Both Novel and Textbook inherit the constructors in Book.
(E) If the Book class has a private method called readFile, this method may not be accessed in either the Novel or Textbook classes.
2. A programmer is designing a program to catalog all books in a library. He plans to have a Book class that stores features of each book: author, title, isOnShelf, and so on, with operations like getAuthor, getTitle, getShelfInfo, and setShelfInfo. Another class, LibraryList, will store an array of Book objects. The LibraryList class will include operations such as listAllBooks, addBook, removeBook, and searchForBook. The programmer plans to implement and test the Book class first, before implementing the LibraryList class. The programmer's plan to write the Book class first is an example of
(A) top-down development.
(B) bottom-up development.
(C) procedural abstraction.
(D) information hiding.
(E) a driver program.
Questions 3–4 refer to the Card and Deck classes shown below.

```java
public class Card {
    private String mySuit;
    private int myValue; //0 to 12

    public Card(String suit, int value) {
        /* implementation */
    }

    public String getSuit() {
        return mySuit;
    }

    public int getValue() {
        return myValue;
    }

    public String toString() {
        String faceValue = "";
        if (myValue == 11)
            faceValue = "J";
        else if (myValue == 12)
            faceValue = "Q";
        else if (myValue == 0)
            faceValue = "K";
        else if (myValue == 1)
            faceValue = "A";
        else if (myValue >= 2 && myValue <= 10)
            return myValue + " of " + mySuit;
        else
            return faceValue + " of " + mySuit;
    }
}

public class Deck {
    private Card[] myDeck;
    public final static int NUMCARDS = 52;

    public Deck() {
        ...

        //Simulate shuffling the deck.
        public void shuffle() {
            ...

        //other methods not shown ...
    }
```
3. Which of the following represents correct /* implementation */ code for the constructor in the Card class?

(A) mySuit = suit;
    myValue = value;
(B) suit = mySuit;
    value = myValue;
(C) Card = new Card(mySuit, myValue);
(D) Card = new Card(suit, value);
(E) mySuit = getSuit();
    myValue = getValue();

4. Consider the implementation of a writeDeck method that is added to the Deck class.

    //Write the cards in myDeck, one per line.
    public void writeDeck()
    {
        /* implementation code */
    }

Which of the following is correct /* implementation code */?

    I System.out.println(myDeck);
    II for (Card card : myDeck)
        System.out.println(card);
    III for (Card card : myDeck)
        System.out.println((String) card);

(A) I only
(B) II only
(C) III only
(D) I and III only
(E) II and III only
5. Refer to the following method that finds the smallest value in an array.

```java
//Precondition: arr is initialized with int values.
//Postcondition: Returns the smallest value in arr.
public static int findMin(int[] arr)
{
    int min = /* somevalue */;
    int index = 0;
    while (index < arr.length)
    {
        if (arr[index] < min)
            min = arr[index];
        index++;
    }
    return min;
}
```

Which replacement(s) for /* somevalue */ will always result in correct execution of the findMin method?

I Integer.MIN_VALUE
II Integer.MAX_VALUE
III arr[0]

(A) I only
(B) II only
(C) III only
(D) I and III only
(E) II and III only
Refer to the following class for Questions 6 and 7.

```java
public class Tester {
    private int[] testArray = {3, 4, 5};

    // Add 1 to n.
    public void increment (int n) {
        n++; } } public void firstTestMethod() {
        for (int i = 0; i < testArray.length; i++) {
            increment(testArray[i]);
            System.out.print(testArray[i] + " ");
        }
    }

    public void secondTestMethod() {
        for (int element : testArray) {
            increment(element);
            System.out.print(element + " ");
        }
    }
}
```

6. What output will be produced by invoking `firstTestMethod` for a `Tester` object?
   (A) 3 4 5  
   (B) 4 5 6  
   (C) 5 6 7  
   (D) 0 0 0  
   (E) No output will be produced. An `ArrayIndexOutOfBoundsException` will be thrown.

7. What output will be produced by invoking `secondTestMethod` for a `Tester` object, assuming that `testArray` contains 3, 4, 5?
   (A) 3 4 5  
   (B) 4 5 6  
   (C) 5 6 7  
   (D) 0 0 0  
   (E) No output will be produced. An `ArrayIndexOutOfBoundsException` will be thrown.
8. Consider the following loop, where $n$ is some positive integer.

```java
for (int i = 0; i < n; i += 2)
{
    if (/* test */)
        /* perform some action */
}
```

In terms of $n$, which Java expression represents the maximum number of times that /* perform some action */ could be executed?

(A) $n / 2$
(B) $(n + 1) / 2$
(C) $n$
(D) $n - 1$
(E) $(n - 1) / 2$

9. A method is to be written to search an array for a value that is larger than a given item and return its index. The problem specification does not indicate what should be returned if there are several such values in the array. Which of the following actions would be best?

(A) The method should be written on the assumption that there is only one value in the array that is larger than the given item.
(B) The method should be written so as to return the index of every occurrence of a larger value.
(C) The specification should be modified to indicate what should be done if there is more than one index of larger values.
(D) The method should be written to output a message if more than one larger value is found.
(E) The method should be written to delete all subsequent larger items after a suitable index is returned.
10. When will method whatIsIt cause a stack overflow (i.e., cause computer memory to be exhausted)?

```java
public static int whatIsIt(int x, int y)
{
    if (x > y)
        return x * y;
    else
        return whatIsIt(x - 1, y);
}
```

(A) Only when \( x < y \)
(B) Only when \( x \leq y \)
(C) Only when \( x > y \)
(D) For all values of \( x \) and \( y \)
(E) The method will never cause a stack overflow.

11. The boolean expression \( a[i] == \max \ || \ ! (\max \ != a[i]) \) can be simplified to

(A) \( a[i] == \max \)
(B) \( a[i] != \max \)
(C) \( a[i] < \max \ || \ a[i] > \max \)
(D) true
(E) false

12. Suppose the characters 0,1,...,8,9,A,B,C,D,E,F are used to represent a hexadecimal (base-16) number. Here A = 10, B = 11, ..., F = 15. What is the largest base-10 integer that can be represented with a two-digit hexadecimal number, such as 14 or 3A?

(A) 32
(B) 225
(C) 255
(D) 256
(E) 272

13. Consider a Clown class that has a default constructor. Suppose a list `ArrayList<Clown> list` is initialized. Which of the following will not cause an `IndexOutOfBoundsException` to be thrown?

(A) for (int i = 0; i <= list.size(); i++)
    list.set(i, new Clown());
(B) list.add(list.size(), new Clown());
(C) Clown c = list.get(list.size());
(D) Clown c = list.remove(list.size());
(E) list.add(-1, new Clown());

GO ON TO THE NEXT PAGE.
Questions 14–16 refer to the Point, Quadrilateral, and Rectangle classes below:

```java
public class Point
{
    private int xCoord;
    private int yCoord;

    //constructor
    public Point(int x, int y)
    {
        ...
    }

    //accessors
    public int get_x()
    {
        ...
    }

    public int get_y()
    {
        ...
    }

    //other methods not shown ...
}

public abstract class Quadrilateral
{
    private String myLabels;  //e.g., "ABCD"

    //constructor
    public Quadrilateral(String labels)
    { myLabels = labels; }

    public String getLabels()
    { return myLabels; }

    public abstract int perimeter();
    public abstract int area();
}
```

GO ON TO THE NEXT PAGE.
public class Rectangle extends Quadrilateral
{
    private Point myTopLeft; //coords of top left corner
    private Point myBotRight; //coords of bottom right corner

    //constructor
    public Rectangle(String labels, Point topLeft, Point botRight)
    { /* implementation code */ }

    public int perimeter()
    { /* implementation not shown */ }

    public int area()
    { /* implementation not shown */ }

    //other methods not shown ...
}

14. Which statement about the Quadrilateral class is false?
   (A) The perimeter and area methods are abstract because there's no suitable
default code for them.
   (B) The getLabels method is not abstract because any subclasses of
       Quadrilateral will have the same code for this method.
   (C) If the Quadrilateral class is used in a program, it must be used as a super-
       class for at least one other class.
   (D) No instances of a Quadrilateral object can be created in a program.
   (E) Any subclasses of the Quadrilateral class must provide implementation
code for the perimeter and area methods.

15. Which represents correct /* implementation code */ for the Rectangle constructor?

   I super(labels);
   II super(labels, topLeft, botRight);
   III super(labels);
      myTopLeft = topLeft;
      myBotRight = botRight;

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

GO ON TO THE NEXT PAGE.
16. Refer to the Parallelogram and Square classes below.

```java
public class Parallelogram extends Quadrilateral {
    //private instance variables and constructor not shown ...

    public int perimeter() {
        /* implementation not shown */
    }

    public int area() {
        /* implementation not shown */
    }
}

public class Square extends Rectangle {
    //private instance variables and constructor not shown ...

    public int perimeter() {
        /* implementation not shown */
    }

    public int area() {
        /* implementation not shown */
    }
}
```

Consider an ArrayList<Quadrilateral> quadList whose elements are of type Rectangle, Parallelogram, or Square.

Refer to the following method, writeAreas:

```java
/* Precondition: quadList contains Rectangle, Parallelogram, or Square objects in an unspecified order. */
public static void writeAreas(List<Quadrilateral> quadList) {
    for (Quadrilateral quad : quadList)
        System.out.println("Area of " + quad.getLabels() + " is " + quad.area());
}
```

What is the effect of executing this method?

(A) The area of each Quadrilateral in quadList will be printed.
(B) A compile-time error will occur, stating that there is no area method in abstract class Quadrilateral.
(C) A compile-time error will occur, stating that there is no getLabels method in classes Rectangle, Parallelogram, or Square.
(D) A NullPointerException will be thrown.
(E) A ClassCastException will be thrown.

GO ON TO THE NEXT PAGE.
17. Refer to the doSomething method:

```java
// postcondition
public static void doSomething(List<SomeType> list, int i, int j)
{
    SomeType temp = list.get(i);
    list.set(i, list.get(j));
    list.set(j, temp);
}
```

Which best describes the postcondition for doSomething?

(A) Removes from list the objects indexed at i and j.
(B) Replaces in list the object indexed at i with the object indexed at j.
(C) Replaces in list the object indexed at j with the object indexed at i.
(D) Replaces in list the objects indexed at i and j with temp.
(E) Interchanges in list the objects indexed at i and j.

18. Consider the NegativeReal class below, which defines a negative real number object.

```java
public class NegativeReal
{
    private Double myNegReal;

    // constructor. Creates a NegativeReal object whose value is num.
    // Precondition: num < 0.
    public NegativeReal(double num)
    { /* implementation not shown */ }

    // Postcondition: Returns the value of this NegativeReal.
    public double getValue()
    { /* implementation not shown */ }

    // Postcondition: Returns this NegativeReal rounded to the nearest integer.
    public int getRounded()
    { /* implementation */ }
}
```

Here are some rounding examples:

<table>
<thead>
<tr>
<th>Negative real number</th>
<th>Rounded to nearest integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.5</td>
<td>-4</td>
</tr>
<tr>
<td>-8.97</td>
<td>-9</td>
</tr>
<tr>
<td>-5.0</td>
<td>-5</td>
</tr>
<tr>
<td>-2.487</td>
<td>-2</td>
</tr>
<tr>
<td>-0.2</td>
<td>0</td>
</tr>
</tbody>
</table>

Which /* implementation */ of getRounded produces the desired postcondition?

(A) return (int) (getValue() - 0.5);
(B) return (int) (getValue() + 0.5);
(C) return (int) getValue();
(D) return (double) (getValue() - 0.5);
(E) return (double) getValue();
19. Consider the following method.

```java
public static void whatsIt(int n)
{
    if (n > 10)
        whatsIt(n / 10);
    System.out.print(n % 10);
}
```

What will be output as a result of the method call `whatsIt(347)`?

(A) 74  
(B) 47  
(C) 734  
(D) 743  
(E) 347

20. A large list of numbers is to be sorted into ascending order. Assuming that a “data movement” is a swap or reassignment of an element, which of the following is a true statement?

(A) If the array is initially sorted in descending order, then insertion sort will be more efficient than selection sort.  
(B) The number of comparisons for selection sort is independent of the initial arrangement of elements.  
(C) The number of comparisons for insertion sort is independent of the initial arrangement of elements.  
(D) The number of data movements in selection sort depends on the initial arrangement of elements.  
(E) The number of data movements in insertion sort is independent of the initial arrangement of elements.
21. Refer to the definitions of ClassOne and ClassTwo below.

```java
public class ClassOne
{
    public void methodOne()
    {
        ...
    }

    //other methods not shown
}

public class ClassTwo extends ClassOne
{
    public void methodTwo()
    {
        ...
    }

    //other methods not shown
}
```

Consider the following declarations in a client class. You may assume that ClassOne and ClassTwo have default constructors.

```java
ClassOne c1 = new ClassOne();
ClassOne c2 = new ClassTwo();
```

Which of the following method calls will cause an error?

I c1.methodTwo();
II c2.methodTwo();
III c2.methodOne();

(A) None
(B) I only
(C) II only
(D) III only
(E) I and II only
22. Consider the code segment

\[
\text{if} \ (n == 1) \\
\quad k++; \\
\text{else if} \ (n == 4) \\
\quad k += 4;
\]

Suppose that the given segment is rewritten in the form

\[
\text{if} \ (/* \text{condition} */) \\
\quad /* \text{assignment statement} */;
\]

Given that \(n\) and \(k\) are integers and that the rewritten code performs the same task as the original code, which of the following could be used as

(1) /* condition */ and (2) /* assignment statement */?

(A) (1) \(n == 1 \land n == 4\) (2) \(k += n\)
(B) (1) \(n == 1 \land n == 4\) (2) \(k += 4\)
(C) (1) \(n == 1 || n == 4\) (2) \(k += 4\)
(D) (1) \(n == 1 || n == 4\) (2) \(k += n\)
(E) (1) \(n == 1 || n == 4\) (2) \(k = n - k\)

23. Which of the following will execute without throwing an exception?

I String \(s = \text{null};\)  
String \(t = "";\)  
if \((s.\text{equals}(t))\)  
\quad \text{System.out.println("empty strings?");}\n
II String \(s = "\text{holy}";\)  
String \(t = "\text{moly}";\)  
if \((s.\text{equals}(t))\)  
\quad \text{System.out.println("holy moly!");}\n
III String \(s = "\text{holy}";\)  
String \(t = s.\text{substring}(4);\)  
\quad \text{System.out.println(s + t);}\n
(A) I only  
(B) II only  
(C) III only  
(D) I and II only  
(E) II and III only
24. Three numbers $a$, $b$, and $c$ are said to be a **Pythagorean Triple** if and only if the sum of the squares of two of the numbers equals the square of the third. A programmer writes a method `isPythTriple` to test if its three parameters form a Pythagorean Triple:

```java
// Returns true if $a \cdot a + b \cdot b = c \cdot c$; otherwise returns false.
public static boolean isPythTriple(double a, double b, double c) {
    double d = Math.sqrt(a * a + b * b);
    return d == c;
}
```

When the method was tested with known Pythagorean Triples, `isPythTriple` sometimes erroneously returned `false`. What was the most likely cause of the error?

(A) Round-off error was caused by calculations with floating-point numbers.
(B) Type `boolean` was not recognized by an obsolete version of Java.
(C) An overflow error was caused by entering numbers that were too large.
(D) $c$ and $d$ should have been cast to integers before testing for equality.
(E) Bad test data were selected.

25. Refer to the following class, containing the **mystery** method.

```java
public class SomeClass {
    private int[] arr;

    // Constructor. Initializes arr to contain nonnegative
    // integers $k$ such that $0 \leq k \leq 9$.
    public SomeClass() {
        /* implementation not shown */
    }

    public int mystery() {
        int value = arr[0];
        for (int i = 1; i < arr.length; i++)
            value = value * 10 + arr[i];
        return value;
    }
}
```

Which best describes what the mystery method does?

(A) It sums the elements of `arr`.
(B) It sums the products $10 \cdot arr[0] + 10 \cdot arr[1] + \ldots + 10 \cdot arr[arr.length-1]$.
(C) It builds an integer of the form $d_1d_2d_3\ldots d_n$, where $d_1 = arr[0]$
(D) It builds an integer of the form $d_1d_2d_3\ldots d_n$, where
    $d_1 = arr[arr.length-1]$, $d_2 = arr[arr.length-2]$, $\ldots$, $d_n = arr[0]$.
(E) It converts the elements of `arr` to base-10.
Questions 26 and 27 refer to the search method in the Searcher class below.

```java
public class Searcher
{
    private int[] arr;

    //Constructor. Initializes arr with integers.
    public Searcher()
    { /* implementation not shown */ }

    /* Precondition: arr[first]...arr[last] sorted in ascending order.
    * Postcondition: Returns index of key in arr. If key not in arr,
    * returns -1. */
    public int search(int first, int last, int key)
    {
        int mid;
        while (first <= last)
        {
            mid = (first + last) / 2;
            if (arr[mid] == key) //found key, exit search
                return mid;
            else if (arr[mid] < key) //key to right of arr[mid]
                first = mid + 1;
            else //key to left of arr[mid]
                last = mid - 1;
        }
        return -1; //key not in list
    }
}
```

26. Which assertion is true just before each execution of the while loop?
(A) arr[first] < key < arr[last]
(B) arr[first] ≤ key ≤ arr[last]
(C) arr[first] < key < arr[last] or key is not in arr
(D) arr[first] ≤ key ≤ arr[last] or key is not in arr
(E) key ≤ arr[first] or key ≥ arr[last] or key is not in arr

27. Consider the array a with values as shown:

4, 7, 19, 25, 36, 37, 50, 100, 101, 205, 220, 271, 306, 321

where 4 is a[0] and 321 is a[13]. Suppose that the search method is called with first = 0 and last = 13 to locate the key 205. How many iterations of the while loop must be made in order to locate it?
(A) 3
(B) 4
(C) 5
(D) 10
(E) 13

GO ON TO THE NEXT PAGE.