Questions 1–3 refer to the Worker class below:

```
public class Worker {
        private String name;
        private double hourlyWage;
        private boolean isUnionMember;
        public Worker()
        { /* implementation not shown */ }

        public Worker(String aName, double anHourlyWage, boolean union)
        { /* implementation not shown */ }

        //Accessors getName, getHourlyWage, getUnionStatus are not shown.

        /** Permanently increase hourly wage by amt.
        * @param amt the amount of wage increase
        */

        public void incrementWage(double amt)
        { /* implementation of incrementWage */ }

        /** Switch value of isUnionMember from true to false and * vice versa. */
        public void changeUnionStatus()
        { /* implementation of changeUnionStatus */ }

}
```

1. Refer to the incrementWage method. Which of the following is a correct /* implementation of incrementWage */?

(A) return hourlyWage + amt;
(B) return getHourlyWage() + amt;
(C) hourlyWage += amt;
(D) getHourlyWage() += amt;
(E) hourlyWage = amt;

2. Consider the method changeUnionStatus.
Which is a correct /* implementation of changeUnionStatus */?

I

```
if (isUnionMember)
        isUnionMember = false;
else
        isUnionMember = true;
```

II

```
isUnionMember = !isUnionMember;
```

III

```
if (isUnionMember) isUnionMember = !isUnionMember;
```

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

3. A client method computePay will return a worker's pay based on the number of hours worked.

```
/** Precondition: Worker w has worked the given number of hours.
* @param w a Worker
* @param hours the number of hours worked
* @return amount of pay for Worker w
*/

public static double computePay(Worker w, double hours)
{ /* code */ }
```

Which replacement for /* code */ is correct?

(A) return hourlyWage * hours;
(B) return getHourlyWage() * hours;
(C) return w.getHourlyWage() * hours;
(D) return w.hourlyWage * hours;
(E) return w.getHourlyWage() * w.hours;

4. What output is produced by the following line of code?

```
System.out.println("\"This is\n very strange\"");
```

(A) \This is\n very strange\
(B) "This is very strange"
(C) This is
very strange
(D) \"This is
very strange\"
(E) "This is
very strange"

5. Refer to the nextIntInRange method below:

```
/** @return a random integer in the range low to high, inclusive */
public int nextIntInRange(int low, int high)
{ return /* expression */ }
```

Which /* expression */ will always return a value that satisfies the postcondition?
(A) (int) (Math.random() * high) + low;
(B) (int) (Math.random() * (high - low)) + low;
(C) (int) (Math.random() * (high - low + 1)) + low;
(D) (int) (Math.random() * (high + low)) + low;
(E) (int) (Math.random() * (high + low - 1)) + low;

6. Consider an array *arr* and an ArrayList *list*. Both arr and list are initialized with string values. Which of the following code segments correctly appends all the strings in arr to the end of list?

I `for (String s : arr) list.add(s);`

II `for (String s : arr) list.add(list.size(), s);`

III `for (int i = 0; i < arr.length; i++) list.add(arr[i]);`

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

7. Consider the following method:

```
/** Precondition: a[0],a[1]...a[n-1] contain integers. */
public static int someMethod(int[] a, int n, int value)
{
        if (n == 0)
                return -1;
        else
        {
                if (a[n-1] == value)
                        return n - 1;
                else
                        return someMethod(a, n - 1, value);
        }
}
```

The method shown is an example of
(A) insertion sort.
(B) mergesort.
(C) selection sort.
(D) binary search.
(E) sequential search.

8. 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

Questions 9–15 refer to the following Date class declaration:

```
public class Date
{
        private int day;
        private int month;
        private int year;

        public Date() //default constructor
        { ... }

        public Date(int mo, int da, int yr) //constructor
        { ... }

        public int month() //returns month of Date
        { ... }

        public int day() //returns day of Date
        { ... }

        public int year() //returns year of Date
        { ... }

        //Returns String representation of Date as "m/d/y", e.g. 4/18/1985.
        public String toString()
        { ... }
}
```

9. Which of the following correctly constructs a Date object in a client class?

(A) Date d = new (2, 13, 1947);
(B) Date d = new Date(2, 13, 1947);
(C) Date d;
    d = new (2, 13, 1947);
(D) Date d;
    d = Date(2, 13, 1947);
(E) Date d = Date(2, 13, 1947);

10. Which of the following will cause an error message?

I
```
Date d1 = new Date(8, 2, 1947);
Date d2 = d1;
```
II
```
Date d1 = null;
Date d2 = d1;
```
III
```
Date d = null;
int x = d.year();
```

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

11. A client program creates a Date object as follows:

```
Date d = new Date(1, 13, 2002);
```

Which of the following subsequent code segments will cause an error?
(A) String s = d.toString();
(B) int x = d.day();
(C) Date e = d;
(D) Date e = new Date(1, 13, 2002);
(E) int y = d.year;

12. Consider the implementation of a write() method that is added to the Date class:

```
/** Write the date in the form m/d/y, for example 2/17/1948. */
public void write()
{ /* implementation code */ }
```

Which of the following could be used as /* implementation code */?

I
```
System.out.println(month + "/" + day + "/" + year);
```
II
```
System.out.println(month() + "/" + day() + "/" + year());
```
III
```
System.out.println(this);
```

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

13. Here is a client program that uses Date objects:

```
public class BirthdayStuff
{
        public static Date findBirthdate()
        { /* code to get birthDate */ return birthDate; }

        public static void main(String[] args)
        { Date d = findBirthdate(); ... }
}
```

Which of the following is a correct replacement for /* code to get birthDate */?

I
```
System.out.println("Enter birthdate: mo, day, yr: ");
int m = IO.readInt(); //read user input
int d = IO.readInt(); //read user input
int y = IO.readInt(); //read user input
Date birthDate = new Date(m, d, y);
```

II
```
System.out.println("Enter birthdate: mo, day, yr: ");
int birthDate.month() = IO.readInt(); //read user input
int birthDate.day() = IO.readInt(); //read user input
int birthDate.year() = IO.readInt(); //read user input
Date birthDate = new Date(birthDate.month(), birthDate.day(), birthDate.year());
```

III
```
System.out.println("Enter birthdate: mo, day, yr: ");
int birthDate.month = IO.readInt(); //read user input
int birthDate.day = IO.readInt(); //read user input
int birthDate.year = IO.readInt(); //read user input
Date birthDate = new Date(birthDate.month, birthDate.day, birthDate.year);
```

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

14. A method in a client program for the Date class has this declaration:

Date d1 = new Date(mo, da, yr);

where mo, da, and yr are previously defined integer variables. The same method now creates a second Date object d2 that is an exact copy of the object d1 refers to. Which of the following code segments will not do this correctly?

I Date d2 = d1;
II Date d2 = new Date(mo, da, yr);
III Date d2 = new Date(d1.month(), d1.day(), d1.year());

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

15. The Date class is modified by adding the following mutator method:

```
public void addYears(int n) //add n years to date
```

Here is part of a poorly coded client program that uses the Date class:

```
public static void addCentury(Date recent, Date old)
{
    old.addYears(100);
    recent = old;
}

public static void main(String[] args) {
    Date oldDate = new Date(1, 13, 1900);
    Date recentDate = null;
    addCentury(recentDate, oldDate);
    ...
}
```

Which will be true after executing this code?
(A) A NullPointerException is thrown.
(B) The oldDate object remains unchanged.
(C) recentDate is a null reference.
(D) recentDate refers to the same object as oldDate.
(E) recentDate refers to a separate object whose contents are the same as those of oldDate.

Use the declarations below for Questions 16–18.

```
public abstract class Solid {
	private String name;

	//constructor
	public Solid(String solidName)
	{ name = solidName; }

	public String getName()
	{ return name; }

	public abstract double volume();
}

public class Sphere extends Solid {
	private double radius;

	//constructor
	public Sphere(String sphereName, double sphereRadius)
	{ super(sphereName); radius = sphereRadius; }

	public double volume()
	{ return (4.0/3.0) * Math.PI * radius * radius * radius; }
}

public class RectangularPrism extends Solid {
	private double length;
	private double width;
	private double height;

	//constructor
	public RectangularPrism(String prismName, double l, double w, double h)
	{
		super(prismName);
		length = l;
		width = w;
		height = h;
	}

	public double volume()
	{ return length * width * height; }
}
```

16. A program that tests these classes has the following declarations and assignments:

```
Solid s1, s2, s3, s4;
s1 = new Solid("blob");
s2 = new Sphere("sphere", 3.8);
s3 = new RectangularPrism("box", 2, 4, 6.5);
s4 = null;
```

How many of the above lines of code are incorrect?
(A) 0
(B) 1
(C) 2
(D) 3
(E) 4

17. Which is false?

(A) If a program has several objects declared as type Solid, the decision about which volume method to call will be resolved at run time.
(B) If the Solid class were modified to provide a default implementation for the volume method, it would no longer need to be an abstract class.
(C) If the Sphere and RectangularPrism classes failed to provide an implementation for the volume method, they would need to be declared as abstract classes.
(D) The fact that there is no reasonable default implementation for the volume method in the Solid class suggests that it should be an abstract method.
(E) Since Solid is abstract and its subclasses are nonabstract, polymorphism no longer applies when these classes are used in a program.

18. Here is a program that prints the volume of a solid:

```
public class SolidMain
{
	/** Output volume of Solid s. */
	public static void printVolume(Solid s)
	{
		System.out.println("Volume = " + s.volume() + " cubic units");
	}

	public static void main(String[] args)
	{
		Solid sol;
		Solid sph = new Sphere("sphere", 4);
		Solid rec = new RectangularPrism("box", 3, 6, 9);
		int flipCoin = (int) (Math.random() * 2); //0 or 1

		if (flipCoin == 0)
			sol = sph;
		else
			sol = rec;

		printVolume(sol);
	}
}
```

Which is a true statement about this program?

(A) It will output the volume of the sphere or box, as intended.

(B) It will output the volume of the default Solid s, which is neither a sphere nor a box.

(C) A ClassCastException will be thrown.

(D) A compile-time error will occur because there is no implementation code for volume in the Solid class.

(E) A run-time error will occur because of parameter type mismatch in the method call printVolume(sol).

19. A program is to simulate plant life under harsh conditions. In the program, plants die randomly according to some probability. Here is part of a Plant class defined in the program.

```
public class Plant
{
        /** probability that plant dies, a real number between 0 and 1 */
        private double probDeath;

        public Plant(double plantProbDeath, < other parameters >)
        {
                probDeath = plantProbDeath;
                < initialization of other instance variables >
        }

        /** Plant lives or dies. */
        public void liveOrDie()
        {
                /* statement to generate random number */
                if (/* test to determine if plant dies */)
                        < code to implement plant's death >
                else
                        < code to make plant continue living >
        }
        //Other variables and methods are not shown.
}
```

Which of the following are correct replacements for
(1) /* statement to generate random number */ and
(2) /* test to determine if plant dies */?

(A) (1) double x = Math.random();
(2) x == probDeath

(B) (1) double x = (int) (Math.random());
(2) x > probDeath

(C) (1) double x = Math.random();
(2) x < probDeath

(D) (1) int x = (int) (Math.random() * 100);
(2) x < (int) probDeath

(E) (1) int x = (int) (Math.random() * 100) + 1;
(2) x == (int) probDeath

20. A program simulates fifty slips of paper, numbered 1 through 50, placed in a bowl for a raffle drawing. Which of the following statements stores in winner a random integer from 1 to 50?

(A) int winner = (int) (Math.random() * 50) + 1;
(B) int winner = (int) (Math.random() * 50);
(C) int winner = (int) (Math.random() * 51);
(D) int winner = (int) (Math.random() * 51) + 1;
(E) int winner = (int) (1 + Math.random() * 49);

21. Consider this method:

```
public static String doSomething(String s)
{
	final String BLANK = " "; //BLANK contains a single space
	String str = ""; //empty string
	String temp;

	for (int i = 0; i < s.length(); i++)
	{
		temp = s.substring(i, i + 1);
		if (!(temp.equals(BLANK)))
			str += temp;
	}

	return str;
}
```

Which of the following is the most precise description of what doSomething does?
(A) It returns s unchanged.
(B) It returns s with all its blanks removed.
(C) It returns a String that is equivalent to s with all its blanks removed.
(D) It returns a String that is an exact copy of s.
(E) It returns a String that contains s.length() blanks.

22. A list of numbers is stored in a sorted array. It is required that the list be maintained in sorted order. This requirement leads to inefficient execution for which of the following processes?

I Summing the five smallest numbers in the list
II Finding the maximum value in the list
III Inserting and deleting numbers

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

23. Which of the following is not necessarily a feature of a robust program?

(A) Does not allow execution to proceed with invalid data
(B) Uses algorithms that give correct answers for extreme data values
(C) Will run on any computer without modification
(D) Will not allow division by zero
(E) Will anticipate the types of errors that users of the program may make

24. Refer to the following code segment. You may assume that arr is an array of int values.

```
int sum = arr[0], i = 0;
while (i < arr.length)
{
        i++;
        sum += arr[i];
}
```

Which of the following will be the result of executing the segment?
(A) Sum of arr[0], arr[1], . . . , arr[arr.length-1] will be stored in sum.
(B) Sum of arr[1], arr[2], . . . , arr[arr.length-1] will be stored in sum.
(C) Sum of arr[0], arr[1], . . . , arr[arr.length] will be stored in sum.
(D) An infinite loop will occur.
(E) A run-time error will occur

25. Consider the following code segment, applied to list, an ArrayList of Integer values.

```
        int len = list.size();
        for (int i = 0; i < len; i++)
        {
                list.add(i + 1, new Integer(i));
                Object x = list.set(i, new Integer(i + 2));
        }
```

If list is initially 6 1 8, what will it be following execution of the code segment?

(A) 2 3 4 2 1 8
(B) 2 3 4 6 2 2 0 1 8
(C) 2 3 4 0 1 2
(D) 2 3 4 6 1 8
(E) 2 3 3 2

26. A two-dimensional array of double, rainfall, will be used to represent the daily rainfall for a given year. In this scheme, rainfall[month][day] represents the amount of rain on the given day and month. For example,

rainfall[1][15] is the amount of rain on Jan. 15
rainfall[12][25] is the amount of rain on Dec. 25

The array can be declared as follows:

double[][] rainfall = new double[13][32];

This creates 13 rows indexed from 0 to 12 and 32 columns indexed from 0 to 31, all initialized to 0.0. Row 0 and column 0 will be ignored. Column 31 in row 4 will be ignored, since April 31 is not a valid day. In years that are not leap years, columns 29, 30, and 31 in row 2 will be ignored since Feb. 29, 30, and 31 are not valid days. Consider the method averageRainfall below:

```
/** Precondition:
* - rainfall is initialized with values representing amounts
*       of rain on all valid days.
* - Invalid days are initialized to 0.0.
* - Feb 29 is not a valid day.
* Postcondition: Returns average rainfall for the year.
*/
public double averageRainfall(double rainfall[][])
{ double total = 0.0; /* more code */ }
```

Which of the following is a correct replacement for /* more code */ so that the postcondition for the method is satisfied?

I
```
for (int month = 1; month < rainfall.length; month++)
        for (int day = 1; day < rainfall[month].length; day++)
                total += rainfall[month][day];
return total / (13 * 32);
```

II
```
for (int month = 1; month < rainfall.length; month++)
        for (int day = 1; day < rainfall[month].length; day++)
                total += rainfall[month][day];
return total / 365;
```

III
```
for (double[] month : rainfall)
        for (double rainAmt : month)
                total += rainAmt;
return total / 365;
```

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

27. Which of the following, when used as the /* body */ of method sum, will enable that method to compute 1 + 2 + ···+ n correctly for any n > 0?

```
/** @param n a positive integer
* @return 1 + 2 + ... + n
*/
public int sum(int n)
{ /* body */ }
```

I
```
return n + sum(n - 1);
```

II
```
if (n == 1)
        return 1;
else
        return n + sum(n - 1);
```

III
```
if (n == 1)
        return 1;
else
        return sum(n) + sum(n - 1);
```

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

28. Refer to the following recursive method.

```
public int mystery(int n)
{
    if (n < 0)
        return 2;
    else
        return mystery(n - 1) + mystery(n - 3);
}
```

What value is returned by the call mystery(3)?
(A) 12
(B) 10
(C) 8
(D) 6
(E) 4

Questions 29 and 30 refer to method t:

```
/** @param n a positive integer */
public int t(int n)
{
        if (n == 1 || n == 2)
                return 2 * n;
        else
                return t(n - 1) - t(n - 2);
}
```

29. What will be returned by t(5)?
(A) 4
(B) 2
(C) 0
(D) −2
(E) −4

30. For the method call t(6), how many calls to t will be made, including the original call?
(A) 6
(B) 7
(C) 11
(D) 15
(E) 25

31. The elements in a long list of integers are roughly sorted in decreasing order. No more than 5 percent of the elements are out of order. Which of the following is a valid reason for using an insertion sort rather than a selection sort to sort this list into decreasing order?

I There will be fewer comparisons of elements for insertion sort.
II There will be fewer changes of position of elements for insertion sort.
III There will be less space required for insertion sort.

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

Questions 32–36 are based on the binSearch method and the private instance variable a for some class:

```
private int[] a;

/** Does binary search for key in array a[0]...a[a.length-1],
* sorted in ascending order.
* @param key the integer value to be found
* Postcondition:
* - index has been returned such that a[index]==key.
* - If key not in a, return -1.
*/
public int binSearch(int key)
{
	int low = 0;
	int high = a.length - 1;
	while (low <= high)
	{
		int mid = (low + high) / 2;
		if (a[mid] == key)
			return mid;
		else if (a[mid] < key)
			low = mid + 1;
		else
			high = mid - 1;
	}

	return -1;
}
```

A binary search will be performed on the following list.

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
4	7	9	11	20	24	30	41

32. To find the key value 27, the search interval after the first pass through the while loop will be

(A) a[0] . . . a[7]
(B) a[5] . . . a[6]
(C) a[4] . . . a[7]
(D) a[2] . . . a[6]
(E) a[6] . . . a[7]

33. How many iterations will be required to determine that 27 is not in the list?

(A) 1
(B) 3
(C) 8
(D) 27
(E) An infinite loop since 27 is not found

34. What will be stored in y after executing the following?

int y = binSearch(4);

(A) 20
(B) 7
(C) 4
(D) 0
(E) -1

35. If the test for the while loop is changed to while (low < high) the binSearch method does not work as intended. Which value in the given list will not be found?

(A) 4
(B) 7
(C) 11
(D) 24
(E) 30

36. For binSearch, which of the following assertions will be true following every iteration of the while loop?

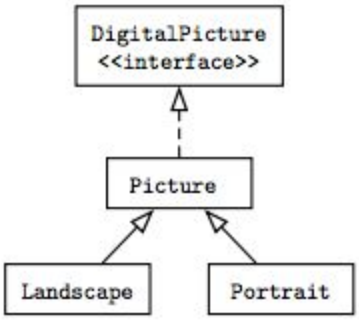
(A) key = a[mid] or key is not in a.
(B) a[low] ≤ key ≤ a[high]
(C) low ≤ mid ≤ high
(D) key = a[mid], or a[low] ≤ key ≤ a[high]
(E) key = a[mid], or a[low] ≤ key ≤ a[high], or key is not in array a.

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

```
String s = "How do you do?";
int index = s.indexOf("o");
while (index >= 0)
{
        System.out.print(index + " ");
        s = s.substring(index + 1);
        index = s.indexOf("o");
}
```

(A) 1 3 2 3
(B) 2 4 3 4
(C) 1 5 8 12
(D) 1 5 8 11
(E) No output because of an IndexOutOfBoundsException

38. Consider a program to manipulate images. The inheritance hierarchy is as follows:



You may assume that Picture has a default (no-argument) constructor, but that Landscape and Portrait do not have any constructors. Which of the following declarations will compile?

I DigitalPicture p = new Portrait();
II Landscape p = new Picture();
III DigitalPicture p = new DigitalPicture();

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

39. Consider a class MatrixStuff that has a private instance variable

```
private int[][] mat;
```

The following method uses a vertical mirror down the center of a matrix to reflect the left half of the matrix onto the right. The following two examples show the result of mirroring a two-dimensional array of numbers from left to right vertically. (Another way of saying this is that the right half of the matrix is replaced by a vertical mirror image of the left half.)

Example 1:

mat	mat after mirroring
1 2 3 4 5	1 2 3 2 1
6 7 8 9 10	6 7 8 7 6
11 12 13 14 15	11 12 13 12 11

Example 2:

mat	mat after mirroring
1 2 3 4	1 2 2 1
5 6 7 8	5 6 6 5
9 10 11 12	9 10 10 9

```
public static void mirrorVerticalLeftToRight(int[][] mat)
{
        int width = mat[0].length;
        int numRows = mat.length;
        for (int row = 0; row < numRows; row++)
                for (int col = 0; col < width/2; col++)
                        /* element assignments */
}
```

Which replacement for /* element assignments */ will make the method work as intended?

(A) mat[row][col] = mat[row][width - col];
(B) mat[row][width - col] = mat[row][col];
(C) mat[row][width - 1 - col] = mat[row][col];
(D) mat[row][col] = mat[row][width - 1 - col];
(E) mat[row][width - 1 - col] = mat[col][row];

40. Consider a square matrix in a class that has a private instance variable mat:

```
private int[][] mat;
```

Method alter in the class changes mat:

```
public void alter()
{
        for (int row = 1; row < mat.length; row++)
                for (int col = 0; col < row; col++)
                        mat[col][row] = mat[row][col];
}
```

If mat has current value

```
{{1, 2, 3},
 {4, 5, 6},
 {7, 8, 9}}
```

what are the contents of mat after method alter has been executed?

(A)
```
{{1, 4, 7},
 {4, 5, 8},
 {7, 8, 9}}
```

(B)
```
{{1, 4, 7},
 {2, 5, 8},
 {3, 6, 9}}
```

(C)
```
{{1, 2, 3},
 {2, 5, 6},
 {3, 6, 9}}
```

(D)
```
{{9, 6, 3},
 {8, 5, 6},
 {7, 8, 9}}
```

(E)
```
{{1, 2, 3},
 {4, 5, 2},
 {7, 4, 1}}
```