1. Whatwilltheoutputbeforthefollowingpoorlyformattedprogramsegment,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.

2. Whichofthefollowingwillevaluatetotrueonlyifbooleanexpressions A,B,and C are all false?

(A) !A && !(B && !C)
(B) !A||!B||!C
(C) !(A||B||C)
(D) !(A&&B&&C)
(E) !A || !(B || !C)

3. Refer to the following code fragment:

```
double answer = 13 / 5;
System.out.println("13 / 5 = " + answer);
```

The output is

```
13 / 5 = 2.0
```

The programmer intends the output to be

```
13 / 5 = 2.6
```

Which of the following replacements for the first line of code will *not* fix the problem?
(A) double answer = (double) 13 / 5;
(B) double answer = 13 / (double) 5;
(C) double answer = 13.0 / 5;
(D) double answer = 13 / 5.0;
(E) double answer = (double) (13 / 5);

4. Supposethatbase-2(binary)numbersandbase-16(hexadecimal)numberscanbe denoted with subscripts, as shown below:

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

Which is equal to $3D_{hex}$?

(A) 111101_{bin}

(B) 101111_{bin}

(C) 10011_{bin}

(D) 110100_{bin}

(E) 101101_{bin}

5. A common use of hexadecimal numerals is to specify colors on web pages. Ev- ery color has a red, green, and blue component. In decimal notation, these are denoted with an ordered triple (*x*,*y*,*z*), where *x*, *y*, and *z* are the three compo- nents, each an int from 0 to 255. For example, a certain shade of red, whose red, green, and blue components are 238, 9, and 63, is represented as (238, 9, 63).

   In hexadecimal, a color is represented in the format #RRGGBB, where RR, GG, and BB are hex values for the red, green, and blue. Using this notation, the color (238, 9, 63) would be coded as #EE093F.

   Which of the following hex codes represents the color (14, 20, 255)?

   (A) #1418FE

   (B) #0E20FE

   (C) #0E14FF

   (D) #0FE5FE

   (E) #0D14FF

6. A client class has a display method that writes the time represented by its parameter:

```
/**
 *  Outputs time t in the form hrs:mins:secs. * @param t the time
*/
public void display (Time t) {
        /* method body */
}
```

Which of the following are correct replacements for /* *method body* */?

```
    I  Time T = new Time(h, m, s);
       System.out.println(T);
    II  System.out.println(t.hrs + ":" + t.mins + ":" + t.secs);
    III  System.out.println(t);
```

(A) Ionly

(B) IIonly

(C) IIIonly

(D) IIandIIIonly

(E) I,II,andIII

Questions 7–13 refer to the following Date class declaration:

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

        public Date() {
                ...
        }

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

        public int month() {
        ... }

        public int day() {
        ... }

        public int year() {
        ... }

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

7. 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);

8. 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

9. 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;

10. 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

11. 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

12. 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

13. 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 Null Pointer Exception is thrown.
(B) The old Date object remains unchanged.
(C) recent Date is a null reference.
(D) recent Date refers to the same object as old Date.
(E) recent Date refers to a separate object whose contents are the same as those of old Date.

14. Here is a program segment to find the quantity base^{exp}. Both base and exp are entered at the keyboard.

```
System.out.println("Enter base and exponent: ");
doublebase=IO.readDouble();       //read user input
double exp = IO.readDouble();     //read user input
/* code to find power, which equals base^exp */
System.out.print(base + " raised to the power " + exp);
System.out.println(" equals " + power);
```

Which is a correct replacement for
/* *code to find* power*, which equals* base^{exp} */?

```
I     double power;
      Math m = new Math(); power = m.pow(base, exp);
II    double power;
      power = Math.pow(base, exp);
III   int power;
      power = Math.pow(base, exp);
```

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

15. Refer to the following code segment. You may assume that array arr1 contains elements arr1[0], arr1[1], . . . , arr1[N-1], where N = arr1.length.

```
int count = 0;
for (int i = 0; i < N; i++)  {
        if (arr1[i] != 0) {
                arr1[count] = arr1[i];
                count++;
        }
}

int[] arr2 = new int[count];
for (int i = 0; i < count; i++) {
        arr2[i] = arr1[i];
}
```

If array arr1 initially contains the elements 0, 6, 0, 4, 0, 0, 2 in this order, what will arr2 contain after execution of the code segment?
(A) 6,4,2
(B) 0,0,0,0,6,4,2
(C) 6,4,2,4,0,0,2
(D) 0,6,0,4,0,0,2
(E) 6,4,2,0,0,0,0

16. Consider writing a program that produces statistics for long lists of numerical data. Which of the following is the best reason to implement each list with an array of int (or double), rather than an ArrayList of Integer (or Double) objects?

(A) Anarray of primitive number types is more efficient to manipulate than an ArrayList of wrapper objects that contain numbers.
(B) Insertion of new elements into a list is easier to code for an array than for an ArrayList.
(C) Removal of elements from a list is easier to code for an array than for an ArrayList.
(D) Accessing individual elements in the middle of a list is easier for an array than for an ArrayList.
(E) AccessingalltheelementsismoreefficientinanarraythaninanArrayList.