AP Computer Science Problem Set 1 **Name:** _____

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. _____ 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);`

3. _____ 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.

4. ____ A program has a `String` variable `fullName` that stores a first name, followed by a space, followed by a last name. There are no spaces in either the first or last names. Here are some examples of `fullName` values: `"Anthony Coppola"`, `"Jimmy Carroll"`, and `"Tom DeWire"`. Consider this code segment that extracts the last name from a `fullName` variable, and stores it in `lastName` with no surrounding blanks:

```
int k = fullName.indexOf(" ");     //find index of blank
String lastName = /* expression */
```

Which is a correct replacement for `/* expression */`?

I `fullName.substring(k);`

II `fullName.substring(k + 1);`

III `fullName.substring(k + 1, fullName.length());`

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

5. ____ This question refers to the following declaration:

```
String line = "Some more silly stuff on strings!";
//the words are separated by a single space
```

What string will `str` refer to after execution of the following?

```
int x = line.indexOf("m");
String str = line.substring(10, 15) + line.substring(25, 25 + x);
```

(A) `"sillyst"`
(B) `"sillystr"`
(C) `"silly st"`
(D) `"silly str"`
(E) `"sillystrin"`

6. ____ 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);`

7. ——— Consider these declarations:

```
String s1 = "crab";
String s2 = new String("crab");
String s3 = s1;
```

Which expression involving these strings evaluates to `true`?

I `s1 == s2`

II `s1.equals(s2)`

III `s3.equals(s2)`

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

Go to the next page

Questions 8 - 10 refer to the following `Date` class declaration.

```
public class Date
{
    private int myDay;
    private int myMonth;
    private int myYear;

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

    public Date(int mo, int day, 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()
    {
        ...
    }
}
```

8. ____ Which of the following correctly constructs a `Date` object?

(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);
```

9. ____ A method in a client program for the `Date` class has this declaration:

```
Date d1 = new Date(month, day, year);
```

where `month`, `day`, and `year` 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(month, day, year);
```

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

10. ____ 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.myYear;
```

Free Response

1. A statistician is studying sequences of numbers obtained by repeatedly tossing a six-sided number cube. On each side of the number cube is a single number in the range of 1 to 6, inclusive, and no number is repeated on the cube. The statistician is particularly interested in runs of numbers. A run occurs when two or more consecutive tosses of the cube produce the same value. For example, in the following sequence of cube tosses, there are runs starting at positions 1, 6, 12, and 14.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Result	1	**5**	**5**	4	3	1	**2**	**2**	**2**	**2**	6	1	**3**	**3**	**5**	**5**	**5**	**5**

The number cube is represented by the following class.

```
public class NumberCube
{
  /** @return an integer value between 1 and 6, inclusive
   */
  public int toss()
  {  /* implementation not shown */  }

  // There may be instance variables, constructors, and methods that are not shown.
}
```

You will implement a method that collects the results of several tosses of a number cube and another method that calculates the longest run found in a sequence of tosses.

(a) Write the method `getCubeTosses` that takes a number cube and a number of tosses as parameters. The method should return an array of the values produced by tossing the number cube the given number of times.

Complete method `getCubeTosses` below.

```
/** Returns an array of the values obtained by tossing a number cube numTosses times.
 *  @param cube a NumberCube
 *  @param numTosses the number of tosses to be recorded
 *         Precondition: numTosses > 0
 *  @return an array of numTosses values
 */
public static int[] getCubeTosses(NumberCube cube, int numTosses)
```

(b) Write the method `getLongestRun` that takes as its parameter an array of integer values representing a series of number cube tosses. The method returns the starting index in the array of a run of maximum size. A run is defined as the repeated occurrence of the same value in two or more consecutive positions in the array.

For example, the following array contains two runs of length 4, one starting at index 6 and another starting at index 14. The method may return either of those starting indexes.

If there are no runs of any value, the method returns `-1`.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Result	1	**5**	**5**	4	3	1	**2**	**2**	**2**	**2**	6	1	**3**	**3**	**5**	**5**	**5**	**5**

Complete method `getLongestRun` below.

```
/** Returns the starting index of a longest run of two or more consecutive repeated values
 *  in the array values.
 *  @param values an array of integer values representing a series of number cube tosses
 *         Precondition: values.length > 0
 *  @return the starting index of a run of maximum size;
 *          -1 if there is no run
 */
public static int getLongestRun(int[] values)
```