August 29 Assignment
AP Computer Science
Victor Kaiser-Pendergrast

Name: _____

Due September 2

Multiple Choice

1. Write the size (in bytes) of each of the following primitive types:

Primitive	**Size in bytes**
int	
float	
double	
boolean	
char	
byte	

2. Given the following method declaration, what is returned by…

```
int someMethod(int x, boolean b) {

        int a = 1;

        if(b) {
                a = 2;
        }

        switch(a * x) {
                case 0:
                case 1:
                case 2:
                        return 3;
                case 4:
                case 5:
                case 6:
                        break;
                default:
                        return x / 2;
        }

        return x * a;
}
```

Method call	**Returned value**
someMethod(0, true);	
someMethod(2, true);	
someMethod(7, false);	

3. What are the values in *arr* after the following code runs?

```
int[] arr = {1, 1, 0, 0, 0};

for (int i = 2; i < arr.length; i++) {
     arr[i] = arr[i-1] + arr[i-2];
}
```

arr's values are { ___, ___, ___, ___, ___ }

4. What is the value of *total* after the following code is executed?

```
int p = 3, q = 1, total = 0;
while (p <= 10)
{
      total += p % q;
      p++;
      q++;
}
```

total is equal to _____

5. What is the value of *num* after this code is run?

```
double x = 5;
double y = 2;

int num = (int)(x + y + x / y - x * y - x / (10 * y));
```

num is equal to _____

6. What is returned by *run( )* ?

```
int f(int x) {
        return x + 2;
}

int g(int x) {
        return x * 2;
}


int run() {
        int x = 1;
        x += f(g(x)) - g(f(x));

        return x;
}
```

run() returns _____

Free Response

1. Implement the following method that returns the minimum of an int array that is passed as an argument:

```
int findMinimum(int[ ] array) {









}
```

2. Write a class that models a rectangular swimming pool. Include the following:
    a. Width, height, depth are attributes of the swimming pool class (assume all units are in feet)
    b. A boolean attribute of the pool of whether or not the pool is filled with water (assume the pool is either completely filled or completely empty)
    c. A default constructor that assigns width, height, and depth reasonable values
    d. A constructor that accepts width, height, and depth as arguments
    e. Getters and Setters for all attributes of the pool
    f. A *amountOfWater( )* method that calculates and returns how much water is in the pool in gallons (note: there are 7.48 gallons in one cubic foot)
    g. A *toString( )* method that describes the pool's state

```
class SwimmingPool {
```

```
}
```