

August 29 Assignment
AP Computer Science
Victor Kaiser-Pendergrast

Name: Victor Kaiser-Pendergrast

Due September 2

Multiple Choice

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

Primitive	Size in bytes
int	4
float	4
double	8
boolean	1
char	1
byte	1

This question was taken from
a real AP exam.

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);	3
someMethod(2, true);	4
someMethod(7, false);	3

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 {1, 1, 2, 3, 5}

This question was taken from an AP
exam and slightly modified.

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 12

P	q	total
3	1	0
4	2	0
5	3	2
6	4	4
7	5	6
8	6	8
9	7	10
10	8	12

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;
}
```

$$\begin{aligned} & 1 + (2+2) - (3*2) \\ & 1 + 4 - 6 \\ & -1 \end{aligned}$$

run() returns -1

questions 4, 5, 6 were from an AP
exam.

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) {
    if (array.length == 0) {
        return 0;
    }
    int min = array[0];
    for (int i = 0; i < array.length; i++) {
        if (array[i] < min) {
            min = array[i];
        }
    }
}
```

2. Write a class that models a rectangular swimming pool. Include the following:

- Width, height, depth are attributes of the swimming pool class (assume all units are in feet)
- 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)
- A default constructor that assigns width, height, and depth reasonable values
- A constructor that accepts width, height, and depth as arguments
- Getters and Setters for all attributes of the pool
- 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)
- A *toString()* method that describes the pool's state

```
class SwimmingPool {
```

```

private double width, height, depth;
// could also use float
public SwimmingPool () {
    width = 10;
    height = 10;
    depth = 5;
}
public SwimmingPool (double width, double height, double depth) {
    this.width = width;
    this.depth = depth;
    this.height = height;
}
public void setWidth (double width) {
    this.width = width;
}
public void setHeight (double height) {
    this.height = height;
}
public void setDepth (double depth) {
    this.depth = depth;
}
public void setFilled (boolean filled) {
    this.filled = filled;
}
public double getWidth () {
    return width;
}
public double getHeight () {
    return height;
}
public double getDepth () {
    return depth;
}

```

```
public boolean getFilled() {  
    return filled;  
}  
public double get amountOfWater() {  
    if (!filled) {  
        return 0;  
    }  
    return width * height * depth * 7.48;  
}  
public String toString() {  
    return "Width:" + width + ", Height:" + height + ", Depth:" +  
        depth + ", Filled:" + filled;  
}  
}
```