

CS171 Pretest  
Name Joseph Barbati

1. What is the output of the following code fragment? 12.

```
int sum = 1;
for (int i = 0; i <= 5; sum = sum + i++);
System.out.print(sum);
```

2. If `a` and `b` are `ints` such that `b != 0` then which of the following expressions is always equivalent to `a%b`? C

- A) `a - (a/b) * b`
- B) `(a/b) * b`
- C) `a - a/b`
- D) `(double)a/b - a/b`

3. Which of the following will always correctly check whether an object variable `obj` contains a null reference? E

- A) `obj.equals(null);`
- B) `null == obj;`
- C) `obj = null;`
- D) `null.equals(obj);`
- E) None of the above

4. A variable, `int x` stores: B

- A) A reference to an `int`
- B) An integer value
- C) The identifier, "`x`"
- D) Lots of goodies for every good Java-slave

5. A variable, `BankAccount x` stores: A

- A) A reference to an object of the `BankAccount` class
- B) An object of the `BankAccount` class
- C) The identifier, "`x`"
- D) Even more goodies than a mere `int x`

6. Consider the following programs and answer the questions.

```
/** a generic employee class */
public class Employee
{
    private String name; // name of the employee
    public Employee (String n) { name = n; }
    public Employee () { name = "Unknown"; }
    public String getName() { return name; }
    public String toString() { return name;}
    public double earnings () { return 0;}
}

/** An hourly employee that makes an earning based on hourly wage */
public class HourlyEmployee extends Employee
{
    private double wage;
    private double hours;

    public HourlyEmployee(String n, double w, double h) {
        super(n); wage = w; hours = h; }

    public double earnings() {
        return wage * hours; }
}

/** A salaried employee that makes a fixed salary */
public class SalariedEmployee extends Employee
{
    private double weeklySalary;

    public SalariedEmployee(String n, double salary) {
        super(n); weeklySalary = salary; }

    public double earnings() {
        return weeklySalary; }
}
```

- a) Identify the superclass and subclass from the above programs.
- b) Name one example each for overloading and overriding from the programs.
- c) Write a method in SalariedEmployee class that overrides the toString() method so that it returns a string that consists of the employee's name and the weekly salary.

a) Employee

b) overloading: Employee constructor

overriding: earnings() method in the Employee class and both subclasses.

c) public String toString() {

    return "Name: " + this.name + "\t Weekly Salary: " + this.weeklySalary;

}

7. Write a method named `inOrder` that takes an array of integers as a parameter and returns the count of elements that are smaller than the element immediately following it. For example, if the array consists of 3, 7, 8, 5, 4, 9, the method should return 3, because  $3 < 7$ ,  $7 < 8$ , and  $4 < 9$ .

```
public static int inOrder(int[] array) {  
    int total = 0;  
    for(int i = 0; i < array.length - 1; i++) {  
        if (array[i] < array[i + 1]) {  
            total++;  
        }  
    }  
    return total;  
}
```

8. When did you take CS170? What do you hope to get out of CS171? Any questions, concerns, suggestions are welcome. Thank you.

Took CS170 in Fall 2015. I hope to gain a deeper understanding of algorithms and data structures, it's pretty much Greek to me at this point (mostly the data structures part). So we shall see what happens.