

For the upcoming midterm, while it may not seem as though you learned much since Midterm 1 you have actually learned a decent bit. On top of the content necessary for midterm 1, since the exams are cumulative it is absolutely fair game for you to be tested on anything up until the lecture prior to the exam.

Here are the main headliner lecture items since the first midterm:

1. Representing Information: Converting from Decimal to Binary/Hex and vice versa
2. Top-Down Design (Cannon's Wishful Programming)
  - a. While it was mentioned briefly you should also know what Bottom-Up Design (Griffin's Constructive Programming) is as well
3. Object-Oriented Design with Multi File Programs
4. Arrays, Statics, Command Line Arguments
5. ArrayLists

Professor Cannon has mentioned that he likes to test on Arrays/ArrayLists so I would be sure to brush up on those.

In terms of how I would prepare for the exam. Understanding the examples from lectures and your homeworks is always a great first place to begin studying. Secondly I would take a look at the Weekly Lecture Notes written by yours truly, as well as the Review Sessions hosted by Gabbie. There is always the textbook which is required reading and anything mentioned in the textbook is fair game to be tested on as well.

I have come up with my own set of practice questions that focus more on the content from this first midterm. I made these on my own so the difficulty may not completely represent how your midterm will go. Assume the code is properly encased unless stated otherwise!

1. What is it called when you create a method with the same name but different formal parameters?

- a. Method Overloading
- b. Method Overriding
- c. Method Aliasing
- d. Method Overwriting
- e. Multithreading

Questions 2-4 Consider the following code snippet:

```
public class BankAccount {  
    public static int bankID = 99;  
    private String name;  
    private double balance;  
    private int userBankID;  
  
    public BankAccount(String name){  
        this.name = name;  
        this.userBankID = bankID++;  
        this.balance = 0.0;  
    }  
  
    //assume proper implementation for the following  
    public void deposit(double amt){}  
    public void withdraw(double amt){}  
    public double getBalance(){}  
    public double getName(){}  
    public int getBankID(){return userBankID;}  
}
```

2. In a Driver Class we instantiate 7 BankAccount objects, how many copies of bankID exist among them?

- a. 0
- b. 7
- c. 3
- d. 4
- e. 1

3. Following from the previous question, if we would call getBankID() on all of these objects which of the following lists represents the values we would get back, assuming we call it on the first object first and so on

- a. 99,99,99,99,99,99,99
- b. 99,101,103,105,107,109,111
- c. 99,100,101,102,103,104,105
- d. 105,105,105,105,105,105,105
- e. 105,104,103,102,101,100,99

4. In the BankAccount class how many accessor methods are there?

- a. 6
- b. 3
- c. 0
- d. 1
- e. 2

Questions 5 and 6 use the following:

$$\text{myNumber}_{10} = 1004$$

5. What is the equivalent of myNumber in Base 16 (Hexadecimal)

- a. AD4
- b. 3EC
- c. 1004
- d. G7A
- e. 1101001

6. What is the equivalent of myNumber in Base 2 (Binary)

- a. 1111101100
- b. 1001001110
- c. 1100110110
- d. 1010101010
- e. 1000101111

7. Consider the following code snippet:

```
int[] myListOfIntegers = new int[5];  
for(int i = 0; i < myListOfIntegers.length; i++)  
    System.out.print(myListOfIntegers[i] + " ");
```

If we were to iterate through myListOfIntegers and print out the results what would be printed?

- a. null null null null null
- b. 1 2 3 4 5
- c. 0 0 0 0 0
- d. A Compiler Error Occurs
- e. A Runtime Error Occurs

8. How many bytes are allocated when we declare a double in Java?

- a. 2 bytes
- b. 4 bytes
- c. 8 bytes
- d. 1 byte
- e. None of the above

9. Consider the following Code Snippet:

```
String names[] = new String[5];  
System.out.println(names[0].contains("a"));
```

What is the result of the above code snippet?

- a. true
- b. false
- c. A Runtime Error occurs
- d. A Compiler Error occurs
- e. None of the above

[LEFT BLANK FOR FORMATTING SCROLL DOWN]

10. Consider the following class:

```
public class StringCollector {  
    public static void main(String[] args){  
        int mystery = 0;  
        for(int i = 0; i < args.length; i++){  
            for(int j = 0; j < args[i].length(); j++){  
                String s1 = args[i].substring(j,j+1);  
                for(int k = 0; k < args[i].length(); k++){  
                    String s2 = args[i].substring(k,k+1);  
                    if(s1.equals(s2)){  
                        mystery++;  
                    }  
                }  
            }  
        }  
        System.out.println(mystery);  
    }  
}
```

What is the value of mystery after I use the following command to execute the above file:

```
javac StringCollector.java && java StringCollector Columbia University
```

- a. 0
- b. 13
- c. 20
- d. A Compiler Error Occurs
- e. A Runtime Error Occurs