**AP Computer Science**  
**Homework Set 1**  
**Fundamentals**

**P1A.** Using “MyFirstApp.java” as a model, write a similar program, “MySecondApp.java”, that prints your “favorites”. Your program should print your food, your favorite color, your favorite movie, and your favorite musical group each on a separate line.

**P1B.** Now let’s practice using an “if statement” and the <, >, and == operators. Write a program that creates two int variables called “a” and “b” and assigns them each an integer values. The program should print a message (using System.out.println) telling the user which if a > b, if a < b, or if a is = to b.

Here’s some sample output if “a” holds the value of 10 and “b” holds the value of 15.

The value of a is 10 and it is less than b whose value is 15.

Here’s some sample output if “a” holds the value of 20 and “b” holds the value of 15.

The value of a is 20 and it is greater than b whose value is 15.

And if the value of a is equal to b, then print:

The value of a is 10 and it is equal to b whose value is 10.
P1C. Write a program that uses an if-else statement that will print whether a number is an even OR an odd number.

Here’s some sample output if the number of 15 (or any odd number):
15 is an odd number

And if the value is 8 (or any even number):
8 is an even number

Note: you can use the “modulus” or “remainder” operator (%) to determine the remainder of a division. For example, if the following lines of code is executed, the value of “remainder1” will be “1”, since 2 goes into 15 seven times with a remainder of 1. However, the value of “remainder2” will be “0” since 2 goes into 16 eight times with a remainder of “0”.

```java
int remainder1 = 15 % 2;  // remainder1 is 1
int remainder2 = 16 % 2;  // remainder2 is 0
int remainder3 = 17 % 2;  // remainder3 is 1
```

The modulus operator is helpful for finding remainders, but it is also the foundation for “public-key cryptography” which allows us to send information such as grades, online purchases, credit card payments, and bank statements securely through the internet.

P1D. Write a program that uses a for loop to print numbers 1 through 10 along with its square and a random number between the number and its square inclusive, each separated by a tab. In Java, you can create a tab by using the escape code “\t” as shown below:

```java
System.out.println( number + "\t" + numberSquared);
```

Here’s some sample output:

```
1  1  1
2  4  2
3  9  5
4 16 11
5 25 22
6 36 32
7 49 49
8 64 55
9 81 56
10 100 67
```

11 is a random number between 4 and 16 (inclusive)

49 is a random number between 7 and 49 (inclusive)
**P1E.** Write a program that creates an array of Strings of your top three favorite movies and an array of Strings of your top three favorite songs. Then use a for loop to print out your three favorite movies on separate lines and a while loop to print out your three favorite songs all on the same line, separated by commas. Leave two blank lines between the movies and the songs.

**P1F.** Write a program that creates an array that can hold 50 integer values. Using a for loop and `Math.random()`, fill the array with random integer values between 0 and 100 inclusive. Use a separate for loop print each of the numbers in the array with a single space between each number.

Note: When using a for loop with arrays, use the variable name “index” as the counter in the loop as shown below:

```java
for( int index = 0, index < myArray.length; index++ )
```

**P1G.** Write a program that creates an array that can hold 10 integer values. Then fill and print the array according the following specifications:

a. Use a for loop and `Math.random()` to fill the array with random integer values between 0 and 100 THAT CAN BE EITHER POSITIVE OR NEGATIVE (hint: randomly multiply each number by “-1”).

b. Use a separate while loop to print each of the numbers in the array on a separate line.

c. The program should then print out the number of positive integers in the array.
P1H. Write a program that creates an array that can hold three Strings. Populate the array with name of your three favorite musical artists other than “Michael Jackson” and print the array using a for loop.

Next, use Math.random() to select a random index in the array in which to place “Michael Jackson”, effectively overwriting one of your three favorite artists. Print the array a second time using a for loop verifying that “Michael Jackson” has arrived.

Use a third for loop to traverse the array and find “Michael Jackson”. Print a statement for each element in the array stating whether or not you found Michael. Below is sample output:

```
No Michael in music[0]
No Michael in music[1]
We found Michael!
```

Check out the LewTube video on how to use the .equals() method of the String class to check for the “equality” of Strings. Be sure to write your code so that if the number of elements in the array is changed, the traversal and “locating” code still works without having to make changes to your code.

The phrase “breaking your code” or “broken code” is used when changes to your data (length of an array or contents of an array among other things) causes incorrect output by your program.
**P11.** Write a program to simulate a login process for a website. Allow the user to input a `String` for the username and an `int` for a password as shown below (we will address more complex passwords later.)

If the username AND password match the correct values, then print the following statement (or a reasonable facsimile of) using `System.out.println()`:

"You have successfully logged in"

If the username/password combination does not match, print the following statement (or a reasonable facsimile of) using `System.out.println()`:

"Login unsuccessful...try again"
By the end of the lesson students should be able to:

a. Type in, compile, and run a Java program.
b. State the basic Java program structure and components (source file, class, and methods).
c. State and describe the two types of errors in programs: logical and syntactical.
d. Debug a Java program with both logical and syntactical errors.
e. Write while and for loops that are syntactically correct.
f. Write if and if-else statements that are syntactically correct.
g. Use the relational operators >, <, >=, <=, ==, and != within the context of an if statement, for loop, and while loop.
h. Use the Boolean operators && and || within the context of if statements and while loops.
i. Write comments using the // and /* …*/, and javadoc constructs.
j. Write the Java code to create a String object.
k. Write the Java code to create an array of String objects or any primitive type.
l. Write the Java code to create an ArrayList of String objects.
m. Write the Java code to access any element in an array using [] notation.
n. Write the Java code to print output using “System.out.println()”.
o. Write code using Math.random() that returns a floating point number (double) from 0 inclusive to 1 exclusive.
p. Write the Java code to create random numbers in any given range using Math.random().