**CMSC 155 Spring 2020**

**Homework 3 (135 points available)**

**DUE: Midnight THURSDAY, Feb. 6**

**Always hand in your code PLUS how you tested it and the results obtained.**

1. Exercises (40 points)
2. **Review/Research:** Write code that declares an int variable, sets it to 1 million, multiplies it by itself, and displays the result. Then repeat with a long variable. EXPLAIN the results. (5 points)

public static void main(String[] args) {  
 int million = 1000000;  
 System.*out*.println(million\*million);

-727379968

public static void main(String[] args) {  
 long million = 1000000;  
 System.*out*.println(million\*million);

1000000000000

The difference is that int doesn’t have enough bits to represent 1million \* 1million while the long variable extends the length of the bits to represent longer numbers.

1. Write and test a **static** **method** that takes 2 strings and an integer n as parameters and **returns** a String consisting of the **1st string concatenated** n **times to the 2nd string**. (10 points)

E.g. “Hi” “Lo” 10 should return “LoHiHiHiHiHiHiHiHiHiHi”

“Lo” “Hi” 5 should return “ HiLoLoLoLoLo”.

1. Write and test a static **method** that takes any 3 strings as parameters and returns true if the **2nd string is strictly between the other 2** **strings** in lexicographic order and false otherwise (Hint use String’s compareTo method). (10 points)

E.g. “hard place” “man” “rock” should return true

(“man” is ‘bigger’ than “hard place” and ‘smaller’ than “rock”)

“Java” “IntelliJ” “IDE” should return true

(“IntelliJ” is ‘smaller’ than “Java” and ‘bigger’ than “IDE”)

“a cup” “a slip” “a lip” should return *false*

(“a slip” is ‘bigger’ than “a cup” but is also bigger than “a lip”)

“this” “this” “that” should return *false*

(“this” is not ‘bigger’ or ‘smaller’ than “this”)

public static boolean letters(String str1, String str2, String str3) {  
 if (str2.compareTo(str1) < 0 && (str2.compareTo(str3) > 0)) {  
 return true;  
 } else if (str2.compareTo(str1) > 0 && (str2.compareTo(str3) < 0)) {  
 return true;  
 } else {  
 return false;  
 }  
}  
  
public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.println("Enter a string: ");  
 String first = input.nextLine();  
 System.*out*.println("Enter a string: ");  
 String second = input.nextLine();  
 System.*out*.println("Enter a string: ");  
 String third = input.nextLine();  
 System.*out*.println(*letters*(first,second,third));  
  
}

1. **Strings vs Booleans**. Explain the advantage of having a method return a boolean (true or false) instead of the strings "True" or "False". (5 points)

The advantage of returning a Boolean rather than a string of “true” or “false” is that when returning a Boolean, we can make so more comparisons rather than if we were writing an if statement to check if two things are true or false then print them. It is much easier to compare with a Boolean.

1. **Print vs Return.** Explain why a method should either return a value or print output, but NOT BOTH. (5 points)

A method can never do both because of how we define the method. If we define it as void, we can’t return anything but if it is defined as a String then it will return a string.

1. Write and test a static method that takes one parameter, an integer. The method should calculate how many numbers between 1 and 100 are evenly divisible by that number using a **while** loop. **Do not simply divide 100 by the number** (10 points)

1. Createa class called Student, with attributes for a student’s name and ID number. Define an appropriate constructor and toString method. We will be building on this class in future weeks.

(10 points)

public class Student {  
   
 private String name;  
 private int id;  
   
 public Student(String name, int id){  
 this.id=id;  
 this.name=name;  
 }  
   
 public String toString() {  
 return "Student " + name + "with ID number " + id;  
 }

1. Createa class called Course, with attributes for a course’s department, course number and credit hours. Define an appropriate constructor and toString method. We will be building on this class in future weeks. (10 points)

public class Course {  
   
 private String dept;  
 private int course;  
 private double credits;  
   
 public Course(String dept, int course, double credits){  
 this.course=course;  
 this.credits=credits;  
 this.dept=dept;  
 }  
   
 public String toString(){  
 return " ";  
 }

1. In Student class create a method that changes the student’s name. It should take one parameter, the new name and it should not return anything. (5 points)

public void changeName(String name){  
 String newName = "Jared";  
 name = newName;  
}

1. Write and run **test code** that instantiates some student and course objects and prints them out. Each student has a name and ID number. Each course has a department (e.g. “CMSC”), course number (e.g. 155) and credit hours (e.g. 4) Also test the method for changing the student’s name. (5 points)

public class StudentCourseTest {  
  
 public static void main(String[] args) {  
 Student stu1 = new Student("Josh",331429);  
 Course course1 = new Course("CMSC",155, 4);  
 System.*out*.println(stu1);  
 System.*out*.println(course1);  
 }

1. The method with nested loops shown below with input parameters 3 and 4: (20 points)

public static void printRows1(int maxRows, int maxCols) {

for(int row = 0; row < maxRows; row++) {

for (int col = 0; col < maxCols; col++) {

System.out.print("\*");

}

System.out.println();

}

}

public static void main(String[] args) {

printRows1(3,4);

}

Prints the following to the console:

\*\*\*\*

\*\*\*\*

\*\*\*\*

Write a method with a single loop that takes the same input parameter and prints the same result. Note: You can only have one loop. Hint: Remember you can use modulo % to determine if one number is evenly divisible by another.