**CMSC 155 Spring 2020**

**Homework 2 (140 points available)**

**DUE: Midnight Thursday Jan 30**

**“There’s an old story about the person who wished his computer were as easy to use as his telephone.  That wish has come true, since I no longer know how to use my telephone.” *- Bjarne Stroustrup***, creator of the programming language C++

Hand in your code, how you tested it and the results obtained.

1. Exercises (40 points)
2. **For Loops.** Predict and verify the output of each of the following. (10 points)
3. String name = "";

**for** (**int** count = 1; count <= 5; count++) name += "hah ";

System.*out*.print(name + "!");

**It will print 5 hah’s with 5 spaces and then an exclamation point.**

**hah hah hah hah hah !**

1. for (int count = 10; count > 0; count--)

System.out.print(count + " ");

**It will print all the numbers less than and equal to 10 with a space in between.**

**10 9 8 7 6 5 4 3 2 1**

1. for (int count = 100; count != 150; count += 5) {

System.out.print(1 + count + " ");

if (count >= 145) System.out.print("Nearly done");

}

**It will print all numbers from 100 to 145 increasing by 5 until count reaches above 145 then it will print “Nearly done”**

**101 106 111 116 121 126 131 136 141 146 Nearly done**

1. for (int count = 20; count > 1; count /= 2)

System.out.print(count + 5 + " ");

System.out.println(count - 5);

System.out.println("Done");

**It will print all numbers greater than one but only as high as count which is 20. It will print count +5 first, then a space followed by count – 5. The next line will print “Done”, followed by the same thing looped until it is less than or equal to 1.**

**25 15**

**Done**

**15 5**

**Done**

**10 0**

**Done**

**7 -3**

**Done**

1. **int** count = 1;

**while** (count < 20) {

System.*out*.println(count); count \*= 2;

}

**This one will print count each loop but multiplied by 2 from the previous count until it reaches 20.**

**1**

**2**

**4**

**8**

**16**

1. **For Loop.** Use a **for** loop to create and complete the following table of Celsius to Fahrenheit temperature conversions from 0 to 100 C). Remember that the formula to convert a temperature in Celsius to Fahrenheit is TempInC \* 9/5 +32. Don’t worry if the formatting is off by a few places. (10 points)

System.*out*.println("Celsius|Fahrenheit");  
System.*out*.println("\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_");  
for (int celsius = 0; celsius <= 100 && celsius >= 0; celsius += 10) {  
 int fahrenheit = celsius \* 9/5 +32;  
 System.*out*.println(" " + celsius +"|" + fahrenheit);  
}

Celsius | Fahrenheit

\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_

0 | 32

10 | 50

20 | 68

… | …

100 | 212

**Celsius|Fahrenheit**

**\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_**

**0|32**

**10|50**

**20|68**

**30|86**

**40|104**

**50|122**

**60|140**

**70|158**

**80|176**

**90|194**

**100|212**

1. **For Loops.** Use a for loop to display the odd numbers from 817 to 11, in that order on the same line separated by spaces. (10 points)

1. **Repeating until a Condition.** A boy and his father push a shopping cart 10 miles the first day. On each following day they push it **70% as far as the day before**. They continue doing this until the distance they travel on a day is less than 0.1 miles; then they stop.

Write a program that shows how far they have travelled at the end of each day. Use a do loop. **Each line** of the output should **display the day number, the distance travelled that day, and the total distance travelled so far**. Distances should be printed to 3 decimal places (See "Displaying Output" slides on Scholar in Intro to Java slides 34 – 35 from week 1). (15 points)

Day 1 distance today = 10.000 total = 10.000

Day 2 distance today = 7.000 total = 17.000

Day 3 distance today = 4.900 total = 21.900

And so on …

The last day, where they travel less than 0.1 miles, should be INCLUDED in the output.

DecimalFormat format = new DecimalFormat("#0.000");  
double travelDistance = 10;  
int day = 1;  
double total = 0;  
do{  
 total += travelDistance;  
 System.*out*.println("Day " + day + " distance today = " + format.format(travelDistance) + " total = " + format.format(total));  
 day++;  
 travelDistance \*= .7;  
} while (travelDistance >= .096);

Day 1 distance today = 10.000 total = 10.000

Day 2 distance today = 7.000 total = 17.000

Day 3 distance today = 4.900 total = 21.900

Day 4 distance today = 3.430 total = 25.330

Day 5 distance today = 2.401 total = 27.731

Day 6 distance today = 1.681 total = 29.412

Day 7 distance today = 1.176 total = 30.588

Day 8 distance today = 0.824 total = 31.412

Day 9 distance today = 0.576 total = 31.988

Day 10 distance today = 0.404 total = 32.392

Day 11 distance today = 0.282 total = 32.674

Day 12 distance today = 0.198 total = 32.872

Day 13 distance today = 0.138 total = 33.010

Day 14 distance today = 0.097 total = 33.107

1. **Changing the For Loop Variable INSIDE the Loop.** What is the output of the following? EXPLAIN why it is bad programming practice to modify the loop variable inside the loop in this way. (5 points)

**for** (**int** count = 1; count < 10; count++)

{

System.*out*.print(count);

**if** (count > 5) count = 10;

}

System.*out*.println();

**The output is “123456”. It is bad programming practice because if you want the if statement to have any effect on the for loop it would need to happen before the print statement otherwise nothing will happen.**

1. **For Loop With Two Variables.** What is the output of the following? EXPLAIN why. (5 points)

**for** (**int** i = 1, j = 4; (i <= 3 && j > 1); i++, j--)

System.*out*.println("i = " + i + " j = " + j);

**The output is i = 1 j = 4**

**i = 2 j = 3**

**i = 3 j = 2**

**This is because each time it loops, it creates a new line but is also adding 1 to I and subtracting 1 from j.**

1. **Nested For Loops**. What is the output of the following? EXPLAIN why. Notice the difference between this and the previous question. (5 points)

**for** (**int** i = 1; i <= 3; i++)

**for (int** j = 4; j > 1; j--)

System.*out*.println("i = " + i + " j = " + j);

**The output is i = 1 j = 4**

**i = 1 j = 3**

**i = 1 j = 2**

**i = 2 j = 4**

**i = 2 j = 3**

**i = 2 j = 2**

**i = 3 j = 4**

**i = 3 j = 3**

**i = 3 j = 2**

**This is because the first for loop contains a nested for loop which runs eveytime j is > 1. When j becomes 1 or less than one, it retracts to the first for loop and adds 1 to i. This reiterates until neither for loop is correct.**

1. **Common Error**. The following is intended to add all the numbers from 1 to 10, but there is an error. EXPLAIN the problem and how to fix it. (5 points)

**int** sum = 0;

**for** (**int** i = 1; i <= 10; i++);

sum = sum + i;

System.*out*.println(sum);

**After plugging this in and trying to run it, I noticed that the for loop does not contain curly braces indicating what is inside the loop. Once I put these curly braces in, I put the sum = sum + i equation into the for loop and was given the sum of 1-10 which is 55.**

1. **String Comparisons.** Suppose s1 is "Simp" and s2 is "son". What are the following? Predict first, then verify. (10 points)
2. **Equality**
   1. (s1 + s2) == "Simpson". 🡪 true 🡪 prints false
   2. (s1 + s2).equals("Simpson") 🡪 true 🡪 prints true
   3. s1 + s2.equals("Simpson") 🡪 false 🡪 prints false
3. **Lexicographic Order**
   1. (s1 + s2).compareTo("Simpson") 🡪 0 🡪 prints 0
   2. "Simpson". compareTo("Timpson") 🡪 -1 🡪 prints -1
   3. "Simpson". compareTo("Silence") 🡪 1 🡪 prints 1
   4. "Simpson". compareTo("Simpleton") 🡪positive 🡪 prints 7
   5. "Simpson". compareTo("Simpsonian") 🡪 negative 🡪 prints -3
   6. "Simpson". compareTo("Sim") 🡪 positive 🡪prints 4
   7. "Simpson". compareTo("simpson") 🡪 negative 🡪 prints -32

1. **Conditionals – switch statement**. Write code that asks the user to enter a letter grade; if it’s a valid letter (A – F) print out its numerical value. (A is 4, B is 3, and so on). If the user’s input is invalid (anything other than A – F), print an appropriate message.

Use the **switch** statement (See “Conditionals” slides on Scholar). (10 points)

Scanner input = new Scanner(System.*in*);  
String grade = input.nextLine();  
if (grade.equals("A") || grade.equals("B")|| grade.equals("C") || grade.equals("D")|| grade.equals("F")) {  
 switch (grade) {  
 case "A":  
 System.*out*.println("4");  
 break;  
 case "B":  
 System.*out*.println("3");  
 break;  
 case "C":  
 System.*out*.println("2");  
 break;  
 case "D":  
 System.*out*.println("1");  
 break;  
 case "F":  
 System.*out*.println("0");  
 break;  
}  
  
} else {  
 System.*out*.println("This input is invalid.");

1. **User Input.** Write code that repeatedly asks the user to enter a word. When the user is done (you need to tell them how to indicate they are done), print out the sum of the lengths of all the words entered. (15 points)