# CS 5000 – Summer 2025
# Assignment #4, 50 Points

## Loops – Chapter 5

**Note:** If you re-upload the files, you must re-upload **ALL** files as the system keeps the most recent uploaded submission only. No zip files!

Develop a complete Java program for each of the following problems. Please name the programs as indicated, add proper program headers, and output labels as shown below. ***Please use only concepts and programming constructs/syntax we discuss to date.***

Make sure you **include a header for each program (see previous assignments).**

**Program #1 (17 points):** Write a Java program, named ***InputSum***, which prompts the user to enter **positive integer numbers** using a sentinel while loop. The program should accept integer inputs until the user enters the value -1 (negative one is the sentinel value that stops the while loop). After the user enters -1, the program should display the entered numbers followed by their sum as shown below. Notice that value -1 is not part of the output. The program should ignore any other negative input and should continue to run.

Make sure the program validates each entered number before processing it as the user may enter negative numbers (other than the sentinel value -1).

**Hint:** to remember/save the entered (good) positive values, you can concatenate them into a string (separated by comas) that you can output later on.

Design your program such that it allows the user to re-run the program with a different set of inputs in the same run (i.e., use a sentinel loop, see example on slide 48, chapter 5). **This characteristic/behavior will be required for future programs, so please make sure you learn how to code sentinel loops in your programs going forward.**

Document your code, organize, and space out your outputs as shown below. The following sample tests show the input prompt and the output labels. Notice the user may enter all values on one line (first test) or one value per line (second test). **DO NOT read inputs as String type**. Make sure your code displays the outputs following the test data format

First test:

```
Enter positive integers (-1 to quit): 10 -5 2 -1

Entered Numbers:  10, 2
The Sum:          12
```

Second test:

```
Enter positive integers (-1 to quit): 1
2
3
4
5
6
-1

Entered Numbers:  1, 2, 3, 4, 5, 6
The Sum:          21
```

Third test:

```
Enter positive integers (-1 to quit): 1 1 -7 1 1 -9 100 -1

Entered Numbers:  1, 1, 1, 1, 100
The Sum:          104
```

Fourth test:

```
Enter positive integers (-1 to quit): 10 -1

Entered Numbers:  10
The Sum:          10
```

Fifth test:

```
Enter positive integers (-1 to quit): -15 -1

Entered Numbers:
The Sum:          0
```

**Program #2 (17 points):** Write a Java program, named ***PasswordTest***, that reads from the user a **string input** (representing a password) and determines whether the password is "Valid Password" or "Invalid Password". A valid password has at least 7 characters and includes at least one lower-case letter, at least one upper-case letter, at least one digit, and at least one character that is neither a letter nor a digit. Your program will need to check each character in the string in order to render a verdict. For example, CS5000/01 is invalid password; however, Cs5000/01 is a valid password. The program should display the entered password followed by the judgment as shown below.

Document your code, organize, and space out your outputs as shown below. Design your program such that it allows the user to re-run the program with different inputs in the same run (i.e., use a sentinel loop structure, see example on slide 48). The following sample tests show only the outputs. Make sure your code displays the outputs following the test data format.

First test:

```
Entered Password:  CS5000/01
Verdict:           Invalid Password
```

Second test:

```
Entered Password:  Cs5000/011
Verdict:           Valid Password
```

Third test:

```
Entered Password:  MyOldDogK9
Verdict:           Invalid Password
```

Fourth test:

```
Entered Password:  MyOldDogK9#
Verdict:           Valid Password
```

Fifth test:

```
Entered Password:  Ab1#
Verdict:           Invalid Password
```

**Program #3 (16 points):** Write a Java program, named ***Pattern***, that uses **nested loops** to print out the following pattern.

```
1 2 3 4 5 6
  1 2 3 4 5
    1 2 3 4
      1 2 3
        1 2
          1
```

Make sure your code displays the outputs following the pattern above.

## Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines posted in D2L. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment due date is posted in D2L.
3. Please compile, run, and test your code right before you upload your java files to the assignment submission folder in D2L.
4. Please upload only the .java files (total 3 files).