

CS101- Algorithms and Programming I

Lab 05

Lab Objectives: `for` and `do-while` loops

- For all labs in CS 101, your solutions must conform to these [CS101 style guidelines](#) (rules!)
 - Create a Lab05 workspace (i.e. the folder `H:\private\cs101\lab05`). This assignment has parts a, b, c, and d, each of which should be placed in a separate project within the same Lab05 workspace. Note: only one project is active at a time. To work (Build/Run) a different project, right click on the project's name and select "Set as active project".
 - You can only use `for` and `do-while` statements for this lab assignment. You cannot use `while` statements. You are not allowed to use arrays or any data structures to store user input.
-

- Create a new project Lab05a. Write a program to generate 1000 random 4-digit numbers. Your program should compare the middle two digits of the number, and display all numbers whose middle two digits are the same. The program should also find and display the following:
 - The highest number generated with same middle digits.
 - The probability of generating a number with same middle digits.
 - The average difference between the middle digits (if not the same middle digits)

Note: You can use the method, `Math.random()` which generates a random double value between 0 and 1 (0 inclusive, 1 exclusive) . You will have to do some calculations to convert the number to a 4-digit int. You are not allowed to use `Math.round()` method.

Sample run (taken with 100 4-digit ints):

```
> run Lab05a
```

```
Numbers with same middle digits:
```

```
6552
```

```
2441
```

```
1008
```

```
2550
```

```
5339
```

```
3990
```

```
3334
```

```
9448
```

```
4229|
```

```
The highest value with same 2nd/3rd digit: 9448
```

```
Probability of generating number with same 2nd/3rd digit: 0.09
```

```
The average difference between the 2nd and 3rd digit: 3.769230769230769
```

```
>
```

- b. Create a new project Lab05b. Write a program that takes an integer, n , and calculate the sum as follows:

$$sum(n) = -1 + \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \dots + \frac{n-1}{n}$$

You should validate the user input as shown in the sample runs.

Sample runs:

```
> run Lab05b
Enter a positive integer: 0
The input n should be positive. Exiting...
> run Lab05b
Enter a positive integer: x
The input n should be an integer. Exiting...
> run Lab05b
Enter a positive integer: 1
The sum is -1.0
> run Lab05b
Enter a positive integer: 2
The sum is -0.5
> run Lab05b
Enter a positive integer: 4
The sum is -0.4166666666666665
> run Lab05b
Enter a positive integer: 10
The sum is -0.3543650793650791
> run Lab05b
Enter a positive integer: 100
The sum is -0.3118278206898044
>
```

- c. Create a new project Lab05c. Your program prompts the user to enter a character, **ch**, and an int value, **height**, and then print out a triangle formed using **ch** characters having a height of **height** characters.

Sample run:

```
> run Lab05c
```

```
Enter character to use:
```

```
Enter height of triangle:
```

```
%%%%%%%%%%
 %%%%%%%%%
  %%%%%%%
   %%%%%
    %%%
     %%
      %
```

- d. Create a new project Lab05d. Write a menu driven program, that asks the user to provide their selection. If any response given other than the “Strings”, “Math”, and “Exit” (case-insensitive), the program should ignore that response and should continue to ask a valid selection.

If the selection is “strings”, it should ask two Strings from user and show the common characters in both Strings. It should be case-insensitive, meaning ‘A’ and ‘a’ should mean the same thing. There should be no duplicates, even if ‘a’ is found more than once, it should be displayed only once.

If the selection is “math”, it should ask the user to provide x, a real value between -1 and 1 (exclusive) , and a precision, a small positive real value less than 1. It should calculate the approximate value of $1 / (1 - x)$ by using the formula given below. Your program stops calculation whenever the increment is less than given precision value.

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + x^5 + \dots \text{ for } -1 < x < 1$$

If the selection is “exit”, the program should say Goodbye and exit.

Sample interactions are shown below.

Sample runs:

```
> run Lab05d
**** Make your selection ****
Strings - Common characters
Math - 1/(1-x) calculation
Exit
Your selection: strings
** Finding Common Characters **
Enter a string: STAR wars
Enter another string: roaring
Common characters: r a
**** Make your selection ****
Strings - Common characters
Math - 1/(1-x) calculation
Exit
Your selection: e
Invalid selection.
**** Make your selection ****
Strings - Common characters
Math - 1/(1-x) calculation
Exit
Your selection: exit
Goodbye!
> |
```

```

> run Lab05d
**** Make your selection ****
Strings - Common characters
Math - 1/(1-x) calculation
Exit
Your selection: 
** 1/(1-x) Calculation **
Enter an x: (-1,1): 
Enter precision: 
Current result is: 1.0
Current result is: 1.5
Current result is: 1.75
Current result is: 1.875
Current result is: 1.9375
Current result is: 1.96875
Current result is: 1.984375
Current result is: 1.9921875
Result is: 1.9921875
**** Make your selection ****
Strings - Common characters
Math - 1/(1-x) calculation
Exit
Your selection: 
** Finding Common Characters **
Enter a string: 
Enter another string: 
Common characters:
**** Make your selection ****
Strings - Common characters
Math - 1/(1-x) calculation
Exit
Your selection: 
Goodbye!
>

```