

# CS101- Algorithms and Programming I

## Lab 05

---

**Lab Objectives:** for, 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 & d, each of which should be placed in a separate project within the same Lab05 workspace.
  - ☐ You can only use "for" and "do-while" loops for this lab assignment. You cannot use "while" or any other repetition method.
- 

- a. Create a new project Lab05a. Write a program that asks the user for an integer N and a character and prints a shape on the screen using the given character and N. In each line, it should print the given character, from starting line 1 to N, and it should increase, i.e line N should have N characters printed, and then it should start to decrease one by one as it goes below.

### Sample run:

```
Enter an integer: 3
Enter a character: *
```

```
*
**
***
**
*
```

### Sample run:

```
Enter an integer: 4
Enter a character: A
```

```
A
AA
AAA
AAAA
AAA
AA
A
```

- b. Create a new project Lab05b. Write a menu driven program, that asks the user to provide an integer from 1 to 3. If any number given out of this range, the program should ignore that number and should continue to ask numbers 1-3.

If number 1 is given, 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 number 2 is given, it should ask the user to provide an integer x and a precision. 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 3 is given, the program should say Goodbye and exit.

### Sample run:

```
=== Make your selection ===
```

```
1) Common characters
2) 1/(1-x) calculation
3) Exit
```

```
Your selection: 1
```

```
=== Common Characters Program ===
```

```
Enter a string: Anakin
Enter another string: Skywalker
Common characters: k a
```

```
=== Make your selection ===
```

```
1) Common characters
2) 1/(1-x) calculation
3) Exit
```

```
Your selection: 2
```

```
=== 1/(1-x) Calculation ===
```

```
Enter an x: (-1,1): -0,3
Enter precision: 0,0001
Current result is: 1.0
Current result is: 0.7
Current result is: 0.7899999999999999
Current result is: 0.7629999999999999
Current result is: 0.7710999999999999
Current result is: 0.7686699999999999
Current result is: 0.7693989999999998
Current result is: 0.7691802999999998
Current result is: 0.7692459099999999
```

Result is: 0.7692459099999999

=== Make your selection ===

- 1) Common characters
- 2)  $1/(1-x)$  calculation
- 3) Exit

Your selection: 3

Goodbye!

- c. Create a new project Lab05c. Write a program that asks the user for a word and displays all substrings of that word, sorted by length.

### Sample run:

```
Enter a word: funny
All substrings of funny, sorted by length are:
funny
funn
unny
fun
unn
nny
fu
un
nn
ny
f
u
n
n
y
```

- d. Create a new project Lab05d. Write a program that reads an integer  $n$  from the user and displays the first  $n$  Fibonacci numbers (5 numbers per line, each number right aligned). The fibonacci numbers are defined as follows:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

### Sample run:

```
Enter a positive integer: -9
Enter a positive integer: -3
Enter a positive integer: 0
Enter a positive integer: 30
30 Fibonacci numbers are:
      1      1      2      3      5
      8     13     21     34     55
     89    144    233    377    610
    987   1597   2584   4181   6765
   10946  17711  28657  46368  75025
  121393 196418 317811 514229 832040
```