EECS168-Lab4

Objectives

- While loop
- do-while loop
- Nested for loop
- Debuggers
- String

String

They can store a series of characters

string word = “Good Morning!”;

You can check their length with the length() method

You can use == and != for comparison

You can look at one specific character using the at() method, e.g.:

```
if (word.at(i)== ‘a’)
Do something;
```

- Don’t forget #include <string>
- Example :

```
for (int i=0; i<=word.length(); i++)
Do something;
```

do while loop review

A *do-while* statement is similar to a *while* statement
except that the loop body is always executed at least **once**.

Syntax of the *do-while* Statement

A Loop Body with Several Statements:

```
do
{
    Statement_1
    Statement_2
      ...
    Statement_Last
} while (Boolean_Expression);
```

body

Do not forget the final semicolon.

A Loop Body with a Single Statement:

```
do
    Statement
while (Boolean_Expression);
```

body

Walter Savitch. 2014. *Problem Solving with C++* (9th ed.). Addison-Wesley Professional.

Nested for loop

- Loop within loop. All loops can be nested. for, while, do-while

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 3; j++)
    {
        cout << "i = " << i << " and j = " << j << "\n";
    }
}
```

```
i = 0 and j = 0
i = 0 and j = 1
i = 0 and j = 2
i = 1 and j = 0
i = 1 and j = 1
i = 1 and j = 2
i = 2 and j = 0
i = 2 and j = 1
i = 2 and j = 2
i = 3 and j = 0
i = 3 and j = 1
i = 3 and j = 2
```

Random numbers

- srand(time(NULL));
  - Used for seeding
    This causes the computer to read its clock to obtain the value for the seed automatically.
    Function time returns the number of seconds that have passed since midnight on January 1, 1970. This value is converted to an unsigned integer and used as the seed to the random number generator.
  - you can call rand() to get back a random positive integer
- Each time you execute this program you will find a different sequence

```
      4     4     3     6     6
      2     6     4     3     3
```

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

- The seed can be given manually, e.g. srand(seed); but we wont do it in this lab

Exercise 1- Guess a Number Game

Create a program, **NumberGuessingGame** that will make the user guess a random number between 1 and 100 (inclusive). The game won't end until the user guesses the correct number, but along the way you will help the user. Until the user guesses the secret number...

•Obtain a guess
•If it's higher than the number, tell the user
•If it's lower than the number, tell the user
•If they guess the number, the game stops and they are prompted to either play again or exit

After they win...

•Congratulate them
•Tell them how many guesses it took
•Tell what their closest incorrect guess was
- If multiple guesses are equally close, display the most recent one

Use **do while** loops in your program.

```
Welcome to the number guessing game.
Guess a number: 45
Sorry, you guess is too low.
Guess a number: 99
Sorry, your guess is too high.
Guess a number: 74
Sorry, your guess is too high.
Guess a number: 72
Sorry, your guess is too low.
Guess a number: 73
You win!
You guessed the secret number after 5 guess(es).
Your closest guess was 72.
Would you like to play again? (y/n): n
```

Exercise 2 - Word game

- Create a new program, **WordGuessingGame**. Using a while loop, create a game for the user. The game requires the user to guess a secret word. The game does not end until the user guesses the word. After they win the game, they are prompted to choose to play again. The secret word that user must guess is "valentine". It is a case-sensitive analysis
- With each guess...
- If the guess does not have an equal number of characters, tell the user if it is too long or too short
- If the guess has the same number of characters, tell the user how many characters are correct
  - The letter must be in the same position to be considered correct

Exercise 3 – Triangle

- Write a program that will print a triangle of asterisks of size **n**, called ***Triangle***. In this program you should use **nested for-loops** (a for loop within another for-loop) and only print out one asterisk out at a time.

```
Enter the number of asterisks for the base of the triangle: 5

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

- Please pay attention to the hint on the wiki page.

Exercise 4 – Pattern Printing

- Create a program called **checker** that will let the user pick the size and symbols for a checkerboard.
- Requirements:
- The board size is nXn
- The size, n, must be greater than 1, so force the user to give you a valid integer
- The user chooses two characters for the two different types of spaces on the board
- You can assume the user will give the correct types of input, but not good values (e.g. for the size n)
- After the user give a valid size and two symbols, print a checkerboard
- After the checkerboard is printed, The chooses when to quit

Exercise for 169

- Update your Word guessing game to check for how letters are correct regardless of length or position.
- You have all the same requirements as before except with each guess the user it told if any of the letters are in the word and how many are in the correct position.
- Note that if a letter is in a guess multiple times you are allowed to count them as separate letters.

Thank you!