# ITSY 1375 Module 1 Assessment Lab - Create a Simple Calculator

## Course Competency(s)

CC1.1 Create a script to solve a problem
LO1.1.1 Compose a flowchart to design a script
LO1.1.2 Utilize programming concepts to create a simple Bash script

## Course Outcomes

CO1: Apply programming design concepts to cyber security administration tasks
CO4: Validate and debug scripts

## NCAE-C CAE-CD - Basic Scripting and Programming (BSP)

BSP.1 Demonstrate proficiency in the use of scripting languages to write simple scripts
BSP.2 Write simple linear and looping scripts.
BSP.3 Write simple and compound conditions within a programming language or similar environment

## Equipment/Supplies Needed

- Windows 10 Education 22H2 or Windows 11 Education 22H2 with Windows Subsystem for Linux 2 (WSL2), Ubuntu 22.04.x LTS

## Assignment

- Create a flowchart in Microsoft Visio
- Write a script that solves a problem statement.

## Procedure

Demonstrate mastery of programming/scripting concepts in BASH include:

- Stdin/Stdout (Input/Output)
- Arithmetic Operators
- Functions
- Flow Control (i.e. if, then, else, case, nested if)
- Operators (and/or/not)

## Problem Statement

**Part 1:**

1. Create a function called header that clears the screen and displays the following information: First and Last name as two separate variables, Class and section (ITSY1375-7P1 for example), the name of the assessment (ITSY 1375 - Module 1 Assessment Lab, and the current date and time.

2. In the main program function, ask a user for their first name and last name, and store the input in two separate variables.

3. Call the header function, passing the first and last names as parameters.

**Part 2:**

1. Ask a user to enter two numbers, and store them in separate variables.

2. Using arithmetic operators, perform the following operations and display the results to the user:
   (1) Addition
   (2) Subtraction
   (3) Multiplication
   (4) Division
   (5) Random Number
      (a) Test which number is larger than the other;

(b) Display a random number between the smaller and larger number.

3. Ask the user if they want to perform the operations again.
    a. If yes, repeat the process.
    b. If no, move on to the next part.

**Part 3:**

1. Ask a user to enter a number between 1 and 10 (inclusive).
2. Using a for loop, display the multiplication table for that number (i.e., 1 x n = n, 2 x n = 2n, 3 x n = 3n, ..., 10 x n = 10n).
3. Ask the user if they want to try another number.
    a. If yes, repeat the process.
    b. If no, move on to the next part.

**Part 4:**

1. Generate a random number between 1 and 100.
2. Using a while loop, display the fibonacci sequence up until the next to last number.
3. Ask the user for the final number in the sequence.
4. If the user guesses the number correctly, congratulate them and ask if they want to play again.
    a. If yes, generate a new random number and start over.
    b. If no, move on to the next part.

**Part 5:**

1. Ask a user to enter a sentence.

2. Using nested conditional statements, determine if the sentence contains any of the following words: "cat", "dog", "bird".

   a. If it does, display a message saying which word(s) were found.

   b. If it doesn't, display a message saying that none of the words were found.

**Part 6:**

1. Ask a user to enter a number between 1 and 10 (inclusive).
2. Using a do...until loop, keep asking the user to guess the number until they get it right.

   a. If the user guesses the number correctly, congratulate them and ask if they want to play again.

   b. If they do, generate a new random number and start over. If they don't, end the program.

## Deliverables

Submit all the following deliverables together:

1. **Flowchart** of the solution using Microsoft Visio.
2. S**creenshots** of the program (3) output with different sets of input values and the expected output for each.
3. **Script**

## Grading Rubric

| Criteria | Exemplary<br>Exceeds<br>Expectations | Accomplished<br>Meets<br>Expectations | Unsatisfactory<br>Does Not<br>Meet Expectations | Score |
|---|---|---|---|---|
| Flowchart | (20) All required elements are present; Clear, understandable, and organized; Symbols are aligned neatly and sized consistently | (15) Less than 2 required elements are missing; Somewhat clear, understandable, and organized; Appropriate symbols are used but sizing not consistent or neat | (0) Missing multiple required elements; No logical flow; Components are out of order or incorrectly applied | |
| Script Elements | 80 | 65 | 0 | |
| Documentation/ Formatting<br>*Proper:*<br>● *Indentation*<br>● *syntax*<br>● *header*<br>● *comments*<br>● *camelCase variable names* | (20) Script is well documented;Header includes all required information; Code is properly implemented; indented/spaced correctly; consistent; Variable names are short, descriptive, and their intent/meaning easy to recognize | (15) Script documentation is lacking in some areas;<br>Header missing required information; Code not always indented/spaced properly;minimal or missing comments where needed; Some or all variables intent or meaning are not clearly defined | Script is missing most or all required documentation; Header is missing most or all required information; Code has no comments; Inconsistent indentation and spacing; Variable names are too long, undescriptive, and/or hard to understand | |
| Execution<br>● Contains no syntax errors<br>● Contains no logic errors | (30) Script executes correctly with no syntax or runtime errors | (25) Script executes partially correct, with some logic errors | (0) Script does not exist or execute correctly due to numerous errors | |
| Requirements<br>● *Test Runs: Documented three test runs with expected output and actual output*<br>● *Screenshots: Submitted 3 screenshots of these tests.*<br>● *Errors: Documented what changes had to be made to correct any errors in the initial test of the program*<br>● *Elements:input/output; Arithmetic; Functions; Operators ;Flow control; Iterative Loops* | (30) 100% requirements are met by Script; Works as, or better than expected; Clearly produces the correct/clear expected results | (25) Some requirements are not or partially implemented by Script;<br>The program partially produces expected results | (0) Several requirements are not implemented by Script; Script produces no results or incorrect/ unexpected results | |
| | | | Total Score | |