Introduction to programming

# Overview:

## Course outline:

* 1. Module 1: Introduction to Core Programming Concepts
  2. Module 2: Core Programming Language Concepts
  3. Module 3: Program Flow
  4. Module 4: Algorithms and Data Structures
  5. Module 5: Error Handling and Debugging
  6. Module 6: Introduction to Object-Oriented Programming
  7. Module 7: More Object-Oriented Programming
  8. Module 8: Introduction to Application Security
  9. Module 9: Core I/O Programming
  10. Module 10: Application Performance and Memory Management

# Requirements:

* Ability to use computers to start programs, open and save files, navigate application menus and interfaces.
* Ability to understand logical concepts such as comparisons.
* Understand number theory.
* Ability to create, understand, and follow structured directions or step-by-step procedures.
* Ability to understand and apply abstract concepts to concrete examples.
* Visual Studio and C# installed.

The following activities are designed to be done in parallel with the course of 10975 Introduction to programming. This course is a starting point to the world of programming. The main objective of this course is to learn the basics of Microsoft Visual Studio and C#. In this course, it is assumed that the students have no prior experience and introduce the concepts needed to progress to the intermediate course on programming. With this in mind, the main objective of these challenges for the students is to start developing the mindset of a developer and to find solution to programming challenges by themselves.

It’s very important to notice that all challenges are connected and intended to be done sequentially. The first challenge is needed to complete the second, and the second is needed to complete the third. In the end, the code should be able to calculate the area, perimeter to given geometric figures. The first challenge’s focus is to get started, consolidate the learning from the module one to five.

With that said, the students will be asked to verify, test and most importantly, write some code to advance in the challenges. But if needed, base files will be provided with errors and missing parts for the students.

Challenge 1:

* Lesson1: Computer Data Storage and Processing
* Lesson2: Application types
* Lesson3: Application Life-Cycle
* Lesson4: Code Compilation

Task:

In this challenge, the goal for the student is to verify and complete the code. In this first challenge, the code is supposed to calculate the Area of three different geometric figures. Triangle, circles, and rectangles. The user interacts with the program via the console and based on the figures the user wishes to calculate, the program will ask for the needed variables.

* Triangle: 3 sides needed.
* Rectangle (or square): 2 sides needed.
* Circle: only radius is needed.

Since this course is not a mathematics course, the equations names needed will be given to the students but not directly implemented in the code, so they will have to search for it and implement it correctly.

Tip:

As stated, for circles, only one variable will be asked because the equation used is the basic one to calculate the area of a circle. For rectangle, two variables will be needed. And for triangles, you will need three variables because you should use the Heron’s formula. But you need to create at least one other variable to be able to choose which figure you wish to calculate.

Texto

Descripción generada automáticamente

Output example:

Since the student will be implementing his own vision of the solution, the output can be different, but the following image shows an example of output for the first challenge.

Texto

Descripción generada automáticamente

Challenge 2:

* Module 6: Introduction to Object-Oriented Programming
* Module 7: More Object-Oriented Programming

Task:

In this challenge, the main objective is to switch from a procedural programming form to Object Oriented Programming. The code will not be improved, the goal here is to learn and reinforce the material learned in creating a real program in OOP such as an abstract class and inheritance.

Texto

Descripción generada automáticamente

Output example:

Texto

Descripción generada automáticamente

Challenge 3:

* Module 8: Introduction to Application Security
* Module 9: Core I/O Programming
* Module 10: Application Performance and Memory Management

Task:

For this last challenge now that the student has an OOP program. The goal will be to level up the capabilities of the program. To do that, students will have to implement the ability to also calculate the perimeter and change the variables input process.

The program should be able to read from a .txt file, all the information needed to calculate chosen figures. Figures will also be selected directly via the .txt file. Since this is a mathematical program, outputs should be in floating points.

Texto

Descripción generada automáticamente

Output example:

Texto

Descripción generada automáticamente