Mohamed Boudiaf University, M'sila

Faculty of Technology

Department of Science and Technology (ST)

Module: Informatics 3

2nd Year ST – 2024/2025

TP1 - Review: Introduction to Program Design

1. Introduction

Programming can be done by intuition or by preparing a work plan. The best approach to a programming project is to apply intuition under the control of a well-designed structure.
The first principle of good programming is persistence. If you are interested in what you are doing, it will not be difficult to develop a passion for sticking with the task, no matter its complexity.
The second principle of good programming is the use of structure-programs is built from the general to the specific. Design is the process of creating something using a work plan. During the design phase, you must have an idea of what the final product will look like. The third principle is the 'divide and conquer' technique.

According to the 14th-century English philosopher William of Ockham, any problem can be divided into part so that solving each part separately will resolve the entire problem.
The design of the Matlab language is mainly guided by this idea. For example, in the C language (or Matlab, in our case), programmers are encouraged to divide the program into a collection of independent functions that are designed and tested separately. This is known as the 'top-down programming' approach.

2. Problem Solving

Since primary school, we have been required to solve problems. Some problems involve calculations, such as those encountered in mathematics or science courses-for example, determining the sum of the first ten odd numbers.
Other problems involve text manipulation, such as identifying Algeria's main resources.
Our goal is to learn how to translate these problem-solving skills into programming so that we can use a computer to handle the mechanical details of solutions. This is the essence of programming: having a problem-solving technique composed of a series of steps that a computer can execute.
When describing a problem-solving technique for a computer, it is necessary to proceed in a very organized manner.

The description must include all the necessary steps to solve the problem in the correct order. Any intermediate information required in one step must be provided in the previous step.

3. Program Design

The design approach used consists of three fundamental steps :
1. Analyze the problem.
2. Create a program profile.
3. Implement the program.
   a. Code the program.
   b. Test and debug the program.
   c. Document the program.

3.1. Problem Analysis

In this step, we must :
- Determine the data to be produced (outputs) as a solution to the problem.
- Determine the necessary input data to produce the outputs.
- Develop an algorithm to obtain the outputs from the inputs in a finite number of steps.
An algorithm is defined as an organized sequence of operations to solve a problem in a finite number of steps.

3.2. Program Profile

In this step, we create a program profile using symbolic statements. Each statement corresponds to a single operation in the program.
For complex programs, we use the top-down design method. The program is divided into tasks, and the task list represents the program profile called the 'main module.'

3.3. Writing the Program

The final phase of program design is writing the source code for the program. In this step, the symbolic language or pseudo-code for the modules is translated into a programming language such as PASCAL, C, or Matlab.
Documentation must be included in the form of comments that describe different sections of the program. Debugging code can also be added to test the program and identify programming errors.
Once the program runs correctly, the debugging code should be removed, but the documentation must remain as a permanent part of the source code for future maintenance.

4. Example of Program Design

Problem:
Write a program that calculates the area of a triangle based on user-specified dimensions.
Problem Analysis:
A triangle is usually specified by two dimensions: height and base. These are the values provided by the user.

Define variables :
- Height = the triangle's height.
- Base = the triangle's base.
- Area = the surface area of the triangle.

Algorithm:
1. Get Height and Base from the user.
2. Calculate the area using the formula: Area = 0.5 * Height * Base.
3. Display the area.

Pseudo-code:

Module 1: Get dimensions
    Display 'Enter the dimensions:'
    Display 'Height = '
    Display 'Base = '
    Enter the value of Height
    Enter the value of Base

Module 2: Calculate the area
Area = 0.5 * Height * Base

Module 3: Display the result**
Display 'The area is'
Display the value of Area