4.2: Program Design Methodologies

IT1406 - Introduction to Programming

Level I - Semester 1

4.2. Program Design Methodologies

4.2.1. Procedure-driven
4.2.2. Event-driven
4.2.3. Data-driven

4.2. Program Design Methodologies

- The fundamental principle of program design is based on the fact that a program accepts input data, processes that data, and then delivers the data to the program user as output. Recently, a number of different approaches to program design have emerged, and the most common are:
  - procedure-driven
  - event-driven
  - data-driven.

4.2. Program Design Methodologies (cont.)

Procedure-driven program design

- The procedure-driven approach to program design is based on the idea that the most important feature of a program is *what* it does – its processes or functions.
- By concentrating on what a program must do, the programmer identifies and organises the processes in the program solution.
- The flow of data into and out of each process or function is then considered and a strategy developed to break each function into smaller and more specific flows of data.
- The details about the actual structure of the data are not considered until all the high-level processes or functions of the program have been defined.

4.2. Program Design Methodologies (cont.)

Event-driven program design

- The event-driven approach to program design is based on the idea that an event or interaction with the outside world can cause a program to change from one known state to another.
- The initial state of a program is identified, then all the triggers that represent valid events for that state are established.
- Each of these events results in the program changing to a new defined state, where it stays until the next event occurs.
- For example, when a program user decides to click the left mouse button, click the right mouse button, drag the mouse or double click the mouse, each action could trigger a different *event* within the program and thus result in a different program state.

4.2. Program Design Methodologies (cont.)

Data-driven program design

- The data-driven approach to program design is based on the idea that the data in a program is more stable than the processes involved.
- It begins with an analysis of the data and the relationships between the data, in order to determine the fundamental data structures.
- Once these data structures have been defined, the required data outputs are examined in order to establish what processes are required to convert the input data to the required output.

4.2. Program Design Methodologies (cont.)

- The choice between procedure-driven, event-driven or data-driven program design methodologies is usually determined by the selection of a programming language.
- However, regardless of the program design method chosen, you must develop the necessary basic skills to be able to design a solution algorithm to a given problem.
- These basic skills include a well defined and disciplined approach to designing the solution algorithm and adherence to the recommended program development process:

4.2. Program Design Methodologies (cont.)

- These basic skills include a well defined and disciplined approach to designing the solution algorithm and adherence to the recommended program development process:

Step 1: Define the problem.
Step 2: Outline the solution (or user interface).
Step 3: Develop the outline into a solution algorithm.
Step 4: Test the algorithm for correctness.
Step 5: Code the algorithm into a specific programming language.
Step 6: Run the program on the computer.
Step 7: Document and maintain the program.

Procedural versus Object-oriented Programming

- Procedural programming is based on a structured, top-down approach to writing effective programs.
- The approach concentrates on *what* a program has to do and involves identifying and organising the *processes* in the program solution.
- The problem is usually broken down into separate tasks or functions and includes top-down development and modular design.

Top-down Development

- In the top-down development of a program design, a general solution to the problem is outlined first.
- This outline is then divided gradually into more detailed steps until finally the most detailed levels have been completed.
- It is only after this process of top-down development (also called *functional decomposition* or *stepwise refinement*) that the programmer starts to code.
- The result of this systematic, disciplined approach to program design is a higher precision of programming than was previously possible.

Modular Design

- Procedural programming also incorporates the concept of modular design, which involves grouping tasks together because they all perform the same function (for example, calculating sales tax or printing report headings).
- Modular design is connected directly to top-down development, as the steps or subtasks into which the program solution is divided actually form the future modules of the program.
- Good modular design also assists in the reading and understanding of the program.

Object-oriented Programming

- Object-oriented programming is also based on breaking down the problem; however, the primary focus is on the *things* (or objects) that make up the program.
- The program is concerned with how the objects behave, so it breaks the problem into a set of separate objects that perform actions and relate to each other.
- These objects have definite properties, and each object is responsible for carrying out a series of related tasks.
- It must be noted, however, that, regardless of design methodology or programming language, all programmers must have the basic skills to design solution algorithms.