4.4: Program Data

IT1406 - Introduction to Programming

Level I - Semester 1

4.4. Program Data

- Because programs are written to process data, you must have a good understanding of the nature and structure of the data being processed.
- Data within a program may be a single variable, such as an integer or a character, or a group item (sometimes called an aggregate), such as an array or a file.

4.4. Program Data

Variables, constants and literals

- A variable is the name given to a collection of memory cells designed to store a particular data item.
- It is called a *variable* because the value stored in those memory cells may change or vary as the program executes. For example, a variable called total_amount may contain several values during the execution of the program.
- A constant is a data item with a name and a value that remain the same during the execution of the program.
- For example, the name *fifty* may be given to a data item that contains the value 50.
- A literal is a constant whose name is the written representation of its value. For example, the data item may contain the literal ‘50’.

4.4. Program Data

Data types

- At the beginning of a program, the programmer must clearly define the form or type of the data to be collected.
- The data types can be elementary data items or data structures.

Elementary data items

- An elementary data item is one containing a single variable that is always treated as a unit.
- These data items are usually classified into data types.
- A data type consists of a set of data values and a set of operations that can be performed on those values.

4.4. Program Data

The most common elementary data types are:

integer:

representing a set of whole numbers, positive, negative or zero

e.g. 3, 576, –5

real:

representing a set of numbers, positive or negative, which may include

values before or after a decimal point. These are sometimes referred to

as floating point numbers

e.g. 19.2, 1.92E+01, –0.01

4.4. Program Data

The most common elementary data types are:

character:

representing the set of characters on the keyboard, plus some special

characters

e.g. ‘A’, ‘b’, ‘$’

Boolean:

representing a control flag or switch that may contain one of only two

possible values, true or false.

4.4. Program Data

Data structures

- A data structure is a structure that is made up of other data items.
- The data items that it contains are its components, which may be elementary data items or another data structure.
- In a data structure, data is grouped together in a particular way, which reflects the situation with which the program is concerned.

4.4. Program Data

The most common data structures are:

record:

a collection of data items or fields that all bear some relationship to one another. For example, a student record may contain the student's number, name, address and enrolled subjects.

file:

a collection of related records. For example, a student file may contain a collection of the above student records.

4.4. Program Data

The most common data structures are:

array:

a data structure that is made up of a number of variables or data items that all have the same data type and are accessed by the same name. For example, an array called *scores* may contain a collection of students' exam scores. Access to the individual items in the array is made by the use of an index or subscript beside the name of the array. For example, scores (3) represents the third score in the array called scores.

string:

a collection of characters that can be fixed or variable. For example, the string *Jenny Parker* may represent a student's name.

4.4. Program Data

Files

- A popular method of storing information is to enter and store data in a file. There are several major advantages of using files:
  - Several different programs can access the same data.
  - The data can be entered and reused several times.
  - The data can be easily updated and maintained.
  - The accuracy of the data is easier to enforce.
- There are two types of files in which data can be stored:
  - sequential or text files, in which data is stored and retrieved sequentially
  - direct or random-access files, in which data is stored and retrieved randomly, using a key or index.
- Sequential files may be opened to read or to write, but not both operations on the same file. Random-access files can be opened to read and write on the same file.

4.4. Program Data

Data validation

- Data should always undergo a validation check before it is processed by a program.
- Different types of data require different checks and can be quite specific; however, the most common data validation checks are as follows:
  - *Correct type:* the input data should match the data type definition stated at the beginning of the program.
  - *Correct range:* the input data should be within a required set of values.
  - *Correct length:* the input data – for example, string – should be the correct
- length.
  - *Completeness:* all required fields should be present.
  - *Correct date:* an incoming date should be acceptable.