

5. Program Design Techniques

Lesson 2: An overview of data and procedures

5.2. Data and Procedures

5.2.1. Data names

- All programming languages have rules when giving names to data items
 - No of characters to be used
 - Type of characters to be used
 - e.g. can we give alphabetic characters , *, \$ etc.
 - Whether they are case sensitive
 - Whether the characters used imply any special characteristics
 - e.g. In the language BASIC, Counter% is an integer, Counter\$ is a string

5.2. Data and Procedures

5.2.2. Explicit vs. implicit definition of data

- Consider
total = total + counter;
- In languages like java **total** and **counter** must already have been defined prior to use. This is known as explicit definition of data.
- In some languages, the compiler is able to “guess” the type of the data by looking at the way it is used. In this example, the compiler can guess that **total** and **counter** are integers because they have been used in an addition operation. This is known as implicit definition of data.

5.2. Data and Procedures

5.2.3. Data types and Data usage

- Data types
 - Numeric data
 - Integers
 - Real Numbers
 - Character data
 - Boolean data

- Data usage
 - Variables
 - Constants

5.2. Data and Procedures

5.2.4. Data structures

- Single items
 - Do not contain sub-items
 - E.g: total, counter
- Structured items
 - group a set of sub-items
 - E.g: a person's name contains a first name, zero or more middle names and a surname
 - Classified under two types
 - Records
 - Arrays

5.2. Data and Procedures

5.2.4. Data structures (cont...)

- Records
 - A structure in which each part has an identifying name, and the data contained in each part of the record may be accessed by referring to that name.
 - For example, a "student record" might be defined as:
 - IdNumber** : integer
 - Name** : string
 - Address** : string
 - DateOf Birth**: string
 - FeePayable** : real.

5.2. Data and Procedures

5.2.4. Data structures (cont...)

- Arrays
 - Consist of sub components which has the same name and are accessed by giving their position within the array
 - The sub components are normally required to be all of the same type and size.
 - For example, the array **DailyRainfall** stores the rainfall for each day of the year
 - **DailyRainfall(365) : real**
 - This array consist of 365 real numbers.

5.2. Data and Procedures

5.2.5. Data scope

- Global
 - Data with Global scope are accessible by a number of procedures.
 - Global data defined within an outer procedure & may then be shared by the inner procedures
 - This removes the need to pass the data from one procedure to another.
 - However, the careless use of global data can result in the data being easily corrupted. Hence, care should be taken when using data with Global scope.
- Local
 - Data with Local scope can be accessed only within the procedure inside which it is defined
 - This removes the possibility of data being corrupted by other procedures
 - It is possible for a number of procedures to each have different variables, each with the identical name

5.2. Data and Procedures

5.2.6. Simple procedures

- A program is a series of procedures designed to process data and produce a result according to a problem specification.