Unit 6 Software design and development

Software structures

Software structure

Software structure is an important design consideration.

It covers:

- how the system will be split into different parts
- good practice when writing programs, e.g.
  - Making the program code easier to read
  - Writing good quality code which is efficient, reliable and easy to use.

Procedures

- When using the procedural approach to programming the system is split into different procedures.
- Each procedure should carry out a **well defined task** within the system.
- In an **event driven** system procedures are written to deal with events, such as the user clicking a button or making a menu selection.

Functions

Most programming languages come with a range of built-in functions which carry out a range of useful general purpose tasks such as:

- validating user input (e.g. checking it is numeric)
- manipulating text (e.g. converting text to upper case)
- generating random numbers
- carrying out common mathematical tasks (e.g. calculating the square root of a number).

Functions

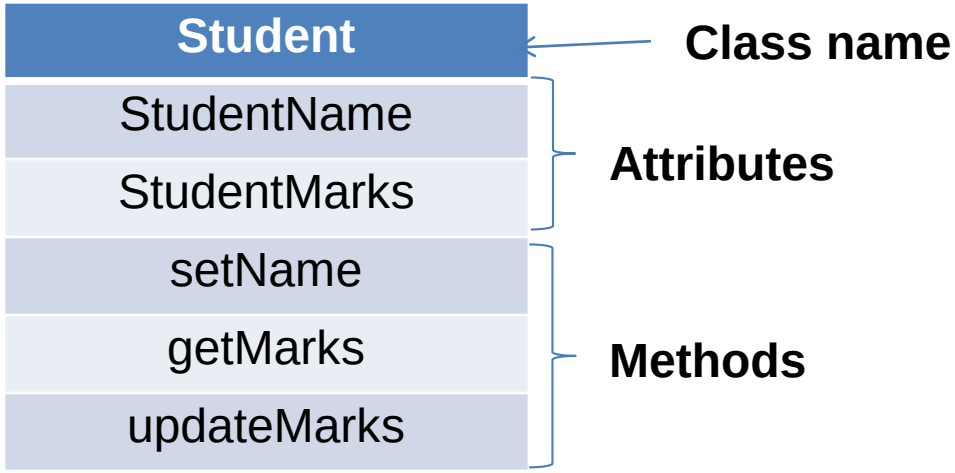
- As well as using the general purpose built in functions programmers can create their own functions.
- These can be used by any procedure within the system.

Objects and classes

- In an object orientated system, it is split into different **classes**.
- A class is a template for an **object** and defines its **attributes** and **methods:**
  - Attributes are the data associated with a class – variables are created within the class to store this data.
  - Methods are the things the class can do – program code is written to implement the methods.

Objects and classes

- Classes can be defined by a **class diagram**, as below.



Objects and classes

- A class is a template for an object, so an actual object for the student class might be:

Student
Sally Smith
102
setName
showMarks
updateMarks

To find out Sally's current mark you would need to use the getMarks method which would return her marks attribute (currently set to 102).