8. Further Programming

20.1 Programming Paradigms

Programming Paradigm

- A set of programming concepts
- Defines the structure and capability of a programming language

Low-Level

- Uses the instruction set of a processor
- Makes use of different addressing modes – immediate, direct, indirect, indexed and relative

Imperative

- Uses variables – changed using assignment statements, rely on an iterative/repetitive method
- Statements provide an explicit sequence of commands for the computer to perform (in the order written) – each line of code changes something in the program being run, commands update the program state

Object Oriented

- Uses the concept of class, inheritance, encapsulation and polymorphism

Object-Oriented Programming Terminology

Properties/Attributes

- The data items or data types defined in a class
- Private attributes
    - ➔ Enforce encapsulation (ensures they are hidden)
    - ➔ Ensure attributes are only accessible using the class's own methods/within the class

Methods

- the modules (e.g. functions) in a class implementing the behaviours that act on the attributes (properties)

Inheritance

- methods and properties contained in one class are reused/made available to a derived class

Polymorphism

- allows methods to be redefined (take on different behaviours) for derived classes

Encapsulation

- process of putting properties and methods inside a class (ensures sensitive data is hidden) together as a single unit

Getter

- method that is used to return the value of a property

Setter

- method that is used to update/set the value of a property

Instance

- an occurrence of an object

Structure/Contents of a Class

- Attributes/properties and their respective data types - these are variables bound to the class
- Methods - subroutines that act upon the attributes
- Getters and Setters - methods that fetch/update the attribute contents
- Constructor - used to create instances of objects in the class

Object vs. Class

- The class is only defined once, but many objects can be created from it - acts as a template from which objects are created
- Objects are allocated memory space when created
- Classes are not allocated memory space when defined
- A class cannot be manipulated, but objects can (a class is not available in memory)
- Classes are defined, objects are created/declared
- Classes can use inheritance, objects cannot

Declarative

- Instructs a program on what needs to be done instead of how to do it – specifies desired result rather than method of acquiring it
- Uses facts, rules and queries to satisfy goals
- Can be logical (states program as set of logical relations) or functional (constructed by applying functions to arguments)

20.2 File Processing and Exception Handling

Exception Handling Routine

- Responds to unexpected events while a program is running – prevents programs from halting unexpectedly
- Traps runtime errors
- Produces meaningful error messages

Exception

- Unexpected event that occurs during the execution of a program
- Causes the program to halt execution
- Examples of Exceptions – end of file, programming/user error, hardware failure, division by zero (runtime errors)