Programming Concepts

Exercises and Solutions

1. What is programming and what is a program?

Programming is writing a sequence of instructions to tell a computer to perform a specific task. A program is a sequence of instructions for a computer.

1. What is the difference between an assembler and a compiler?

An assembler translates the instructions written in an assembly language into a machine language. A compiler translates programs written in a high-level programming language into a machine language.

1. What is machine language and what does a program written in machine language consist of?

Machine language is a set of instructions in binary format, only format computers understand. Machine language program consists of binary sequences of 0s and 1s.

1. What is assembly language and what does a program written in assembly language consists of?

An assembly language is a set of notations to write instructions. It is bit easier to write, read, and understand when compared to binary sequences in machine language. An assembly language program consists of mnemonics.

1. Name three higher-level programming languages.

Java, Scala, SQL

1. Based on the generation of a programming language (1GL, 2GL, etc.), in what categories Java and SQL fall?

Java is 3GL, SQL is 4GL.

1. What is a programming paradigm? Describe procedural, functional, and object-oriented paradigms with an example.

Refer to section ‘Programming Paradigms’ of this chapter in the book.

1. Name the four pillars of the object-oriented programming and describe each of them with an example.

Abstraction: Process of exposing only the essential details of an entity or process.

Encapsulation: Process of bundling data and operations on the data together in an entity

Inheritance: Establishing a parent-child relationship by deriving a new type from an existing type.

Polymorphism: Ability of an entity to take on different meanings in different contexts

Refer to section ‘The Object-Oriented Paradigm and Java’ for examples.

1. What is "true" polymorphism and how does Java support it?

In “true” or universal polymorphism, the same code works for all types. There are two categories: Inclusion polymorphism and Parametric polymorphism.

Java supports inclusion polymorphism using inheritance, which is a subclassing mechanism and parametric polymorphism through generics.

1. What is an abstract data type? How does Java support abstract data type?

An abstract data type is data type where the storage representation of data objects is hidden from users of data type. ADT is defined solely in terms of operations that can be applied to the data objects of its type without knowing the internal representation of the data.

Java supports ADT by providing features to hide data in constructs like class.