**Chapter 1**

Exercise 1.1

1. Programs
2. Input Unit, Output Unit, Memory Unit, Arithmetic and Logic Unit (ALU), Control Unit
3. Machine Language, Assembly Language, High-level Language
4. Compilers
5. Android
6. Open-source software
7. Accelerometer

Exercise 1.2

1. Java
2. Javac
3. .java
4. .class
5. Bytecode

Exercise 1.3

1. Encapsulation
2. Classes
3. Object-oriented analysis and design (OOAD)
4. Inheritance
5. Unified Modeling Language (UML)
6. Attributes

Exercise 1.4

1. Input unit
2. Programming
3. Assembly Language
4. Output Unit
5. Primary Storage (RAM) and Secondary Storage (hard drive, SSD, etc.)
6. Arithmetic logic unit
7. Control unit
8. High-level programming
9. Machine language
10. Control unit

Exercise 1.5

1. Java
2. C
3. Transmission Control Protocol
4. C++

Exercise 1.6

1. Edit, compile, load, verify, and execute
2. Integrated Development Environment (IDE)
3. Java Virtual Machine (JVM)
4. Virtual Machine
5. Class loader
6. Bytecode verifier

Exercise 1.7

1. Compilation Phase (Source Code to Bytecode): The Java compiler (java c) translates human-readable Java source code (.java files) into bytecode (.class files). Bytecode is an intermediate, platform-independent representation of the program. This allows Java Virtual Machine (JVM).
2. Execution Phase (Bytecode to Machine Code): The Java Virtual Machine (JVM) loads the .class file and interprets or compiles it further into machine-specific code. The Just-In-Time (JIT) compiler converts bytecode into native machine code for better performance. This allows Java programs to run efficiently on different platforms without needing recompilation.

Exercise 1.8

1. Object: A wristwatch is an instance of a general category of timekeeping devices.
2. Attributes: It has properties such as, Brand, Color, Size, Type (analog or digital), battery level, Current time
3. Behaviors: It can perform functions such as, displaying time, setting an alarm, showing date, measuring elapsed time (stopwatch feature), adjusting brightness (in smartwatches)
4. Class: A ‘watch’ class might define common properties and methods that all watches share. Specific watches like digital and analog watches would be objects (instances) of this class.
5. Inheritance: A wristwatch can inherit properties from a general Clock class.
6. Modeling: The process of designing a wristwatch in software involves creating an abstraction of its real-world characteristics. This includes defining its attributes, behaviors, and relationships with other objects.
7. Messages: In OOP, objects communicate via messages (method calls). A user pressing the “set-time” button sends a message to the watch to update its time. An alarm goes off, sending a signal to the buzzer to produce sound.
8. Encapsulation: The internal workings of the watch (such as the timekeeping mechanism) are hidden from the user. The user interacts with it through buttons or touchscreen, without needing to understand how it calculates time.
9. Interface: A watch provides interface that allows users to interact with it-buttons, a touchscreen, or even voice commands (in smartwatches). This hides complex internal logic from the user
10. Information Hiding: A watch keeps certain details private, such as the internal algorithm for time adjustment or power management. The user can only access certain functionalities without modifying internal processes.