**Chapter 1**

**Self-Review Exercises 1.1**

**Fill in the blanks in each of the following statements:**

a) Computers process data under the control of sets of instructions called programs.

b) The key logical units of the computer are the arithmetic logic unit,control unit, memory, input unit, output unit, and storage unit.

c) The three types of languages are machine language,assembly language, and high-level language.

d) The programs that translate high-level language programs into machine language are called compilers.

e) Android is an operating system for mobile devices based on the Linux kernel and Java.

f) Release software is generally feature complete, (supposedly) bug free and ready for use by the community.

g) The Wii Remote, as well as many smartphones, use an accelerometer which allows the device to respond to motion.

**1.2 Fill in the blanks in each of the following sentences about the Java environment:**

a) The java command from the JDK executes a Java application.

b) The javac command from the JDK compiles a Java program.

c) A Java source code file must end with the .Java file extension.

d) When a Java program is compiled, the file produced by the compiler ends with the .class file extension.

e) The file produced by the Java compiler contains bytecode that is executed by the Java Virtual Machine.

**1.3 Fill in the blanks in each of the following statements**

a) Objects enable the design practice of information hiding — although they may know how to communicate with one another across well-defined interfaces, they normally are not allowed to know how other objects are implemented.

b) Java programmers concentrate on creating objects, which contain fields and the set of methods that manipulate those fields and provide services to clients.

c) The process of analyzing and designing a system from an object‐oriented point of view is called object‐oriented analysis and design.

d) A new class of objects can be created conveniently by inheritance — the new class (called the subclass) starts with the characteristics of an existing class (called the superclass), possibly customizing them and adding unique characteristics of its own.

e) UML is a graphical language that allows people who design software systems to use an industry‐standard notation to represent them.

f) The size, shape, color, and weight of an object are considered attributes of the object’s class.

**Exercises 1.4 Fill in the blanks in each of the following statements:**

a) The logical unit that receives information from outside the computer for use by the computer is the input unit.

b) The process of instructing the computer to solve a problem is called programming.

c) Assembly language is a type of computer language that uses English-like abbreviations for machine-language instructions.

d) Output unit is a logical unit that sends information which has already been processed by the computer to various devices so that it may be used outside the computer.

e) Primary memory and secondary memory are logical units of the computer that retain information.

f) Arithmetic Logic Unit (ALU) is a logical unit of the computer that performs calculations.

g) Control Unit is a logical unit of the computer that makes logical decisions.

h) High-level languages are most convenient to the programmer for writing programs quickly and easily.

i) The only language a computer can directly understand is that computer’s machine language.

j) Central Processing Unit (CPU) is a logical unit of the computer that coordinates the activities of all the other logical units.

**1.5 Fill in the blanks in each of the following statements:**

a) The Java programming language is now used to develop large-scale enterprise applications, to enhance the functionality of web servers, to provide applications for consumer devices and for many other purposes.

b) C initially became widely known as the development language of the UNIX operating system.

c) The Transmission Control Protocol (TCP) ensures that messages, consisting of sequentially numbered pieces called bytes, were properly routed from sender to receiver, arrived intact and were assembled in the correct order.

d) The C++ programming language was developed by Bjarne Stroustrup in the early 1980s at Bell Laboratories.

1.6 Fill in the blanks in each of the following statements:

a) Java programs normally go through five phases — editing, compiling, linking, loading and executing.

b) A(n) integrated development environment (IDE) provides many tools that support the software development process, such as editors for writing and editing programs, debuggers for locating logic errors in programs, and many other features.

c) The command Java invokes the Java Virtual Machine (JVM), which executes Java programs.

d) A(n) virtual machine is a software application that simulates a computer, but hides the underlying operating system and hardware from the programs that interact with it.

e) The loader takes the .class files containing the program’s bytecodes and transfers them to primary memory.

f) The verifier examines bytecodes to ensure that they’re valid.

1.7 Explain the two compilation phases of Java programs

Java programs are processed in two main phases before they are executed:

1. Compilation Phase (Static Compilation):

   - The Java compiler (`javac`) reads the Java source code (files ending in .java) and checks it for syntax errors.

   - It then translates the source code into an intermediate, platform-independent form called bytecode, which is stored in .class files.

   - This phase ensures that the program is syntactically correct and that it adheres to the Java language rules.

2. Execution Phase (Dynamic Compilation/Interpretation):

   - The Java Virtual Machine (JVM) loads the bytecode from the .class files.

   - Before execution, the JVM verifies the bytecode to ensure it is safe and adheres to Java’s security constraints.

   - The JVM then either interprets the bytecode or, more commonly, uses a \*\*Just-In-Time (JIT) compiler to compile frequently executed parts into native machine code for faster execution.

**1.8 Using a Wrist Watch to Illustrate Object-Oriented Concepts**

A wrist watch can be used as an analogy to explain several object-oriented (OO) concepts:

- Object:

  A watch is a concrete instance of an object. In OO programming, an object is an instance of a class that has a state and behavior. For example, your personal wrist watch is one specific object among many possible watches.

- Attributes:

  Attributes are the properties or characteristics of an object. For a watch, attributes might include the time displayed, brand, color, shape, weight, battery status, and material. These describe what the watch is like.

- Behaviors:

  Behaviors are the actions or functions an object can perform. A watch’s behaviors include displaying time, starting a stopwatch, triggering an alarm, and possibly adjusting time settings. These are analogous to methods in a class.

- Class:

  A class is a blueprint that defines the attributes and behaviors common to all objects of that type. The Watch class could define the properties (attributes like time, brand, etc.) and methods (behaviors like displayTime(), setAlarm(), etc.) that every watch object will have.

- Inheritance:

  Inheritance allows a new class (subclass) to inherit attributes and behaviors from an existing class (superclass). For example, an Alarm Clock can be a subclass of the Watch class. It inherits all the basic functionalities of a watch (like time display) but also adds extra behaviors, such as sounding an alarm.

- Modeling:

  Modeling is the process of representing a real-world object in a computer program. When you model a wrist watch in a program, you abstract its essential features (attributes and behaviors) into a class, leaving out unnecessary details.

- Messages:

  In OO systems, objects communicate by sending messages to each other, which are essentially method calls. For example, when you press a button on your watch, it sends a message to the internal processor to update the time or activate the alarm.

- Encapsulation:

  Encapsulation is the bundling of data (attributes) and methods (behaviors) into a single unit (class) and restricting access to the internal details. In a watch, you interact with the visible display and buttons without needing to know how the internal circuitry works. The inner workings are hidden, and only the necessary interface is exposed.

- Interface:

  An interface defines a contract of methods that a class must implement. For a watch, the interface might be the set of controls (buttons, touch screen) and display that allow the user to interact with the watch. In programming, a Java interface would specify methods (like setTime() or triggerAlarm()) that any implementing class must provide.

- Information Hiding:

  Information hiding is a principle where the internal implementation details of an object are hidden from the outside world. With a watch, the user does not see the complex mechanics or electronic circuits that keep time; they only interact with the simple display and controls. Similarly, in an object-oriented program, the internal data and implementation details are hidden (often marked as private) so that only a defined interface is exposed to the outside world.

**Making a Difference**

**1.9 (Test-Drive: Carbon Footprint Calculator)**

Answer

Carbon footprint calculators estimate the total amount of carbon dioxide (CO₂) and other greenhouse gases that result from an individual’s or organization’s activities—typically measured in metric tons of CO₂ equivalent per year. To do this, they combine the emissions from several sources. Here are some of the key formulas and concepts used:

1. Electricity Consumption:

   Formula:

     Electricity Emissions = Total Electricity (kWh) x Emission Factor (tons CO₂/kWh)

   - Example:

     If you consume 10,000 kWh in a year and the emission factor is 0.0005 tons CO₂ per kWh (a value that varies by region and energy mix), then:

     10,000 x  0.0005 = 5  tons CO₂

2. Transportation (Gasoline Use):

   - Formula:

     Transportation Emissions = Total Miles Traveled\Miles per Gallon (MPG) x Gasoline Emission Factor (tons CO₂/gallon)

   - Example:

     If you drive 15,000 miles a year in a car with 25 MPG, then you use about 600 gallons of gasoline. With an emission factor of about 0.00889 tons CO₂ per gallon:

     600 x  0.00889 =  5.33 tons CO₂

3. Natural Gas (Heating):

   - Formula:

     Heating Emissions = Natural Gas Consumption (therms) x Emission Factor (tons CO₂/therm)

   - Example:

     If you use 500 therms in a year and the emission factor is about 0.0053 tons CO₂ per therm:

     500 x  0.0053 = 2.65 tons CO₂

4. Total Carbon Footprint:

The calculator adds up the emissions from all the sources:

Total Footprint = Electricity Emissions + Transportation Emissions + Heating Emissions +....

This sum gives you your annual carbon footprint in metric tons of CO₂ equivalent.

**1.10 (Test-Drive: Body Mass Index Calculator)**

Obesity causes significant increases in illnesses such as diabetes and heart disease. To determine whether a person is overweight or obese, you can use a measure called the body mass index (BMI). The United States Department of Health and Human Services provides a BMI calculator at [http://www.nhlbi.nih.gov/guidelines/obesity/BMI/bmicalc.htm](http://www.nhlbi.nih.gov/guidelines/obesity/BMI/bmicalc.htm). Use it to calculate your own BMI. A forthcoming exercise will ask you to program your own BMI calculator. To prepare for this, use the web to research the formulas for calculating BMI.

My Calculation:

1. Convert Height to Meters:

   - My height is 5 feet 7 inches.

   - First, convert feet to inches:

     5  ft x 12 = 60 inches

     Add the extra 7 inches: 60 + 7 = 67 inches

   - Convert inches to centimeters (1 inch = 2.54 cm):

     67 x 2.54 = 170.18 cm

   - Convert centimeters to meters:

    170.18  cm/ 100 = 1.7018  m

2. Calculate BMI Using the Metric Formula:

   The BMI formula in metric units is:

   BMI = Weight (kg)\Height (m)^2

   My weight is 50 kg.

   Plug in the values:

BMI = 50/1.7018)^2 = 50/ 2.896 = 17.3

3. Interpretation:

   Underweight: BMI < 18.5

    Normal weight: 18.5 ≤ BMI < 25

   Overweight: 25 ≤ BMI < 30

   Obese: BMI ≥ 30

 With a BMI of approximately 17.3, this indicates that I am in the underweight category.

**1.11 (Attributes of Hybrid Vehicles)**

Hybrid vehicles are increasingly popular because they typically achieve much better mileage than purely gasoline-powered cars. After browsing various websites and studying popular hybrid models (such as the Toyota Prius, Honda Insight, Ford Fusion Hybrid, Hyundai Ioniq Hybrid, and Kia Niro), many hybrid-related attributes can be identified. These attributes fall into several key categories:

1. Fuel Economy:

   - City Miles Per Gallon (MPG):

     Many hybrids advertise high efficiency in urban stop-and-go driving. For example, a Toyota Prius might achieve around 54 MPG in the city.

   - Highway MPG:

     Hybrids also offer impressive highway mileage (often in the high 40s or low 50s MPG).

   - Combined MPG:

     This is a weighted average of city and highway MPG, providing an overall measure of efficiency.

2. Powertrain and Drivetrain Features:

   - Gasoline Engine Specifications:

     Typically, hybrids use a small, efficient internal combustion engine (often a 1.5L–2.0L 4-cylinder) designed to work in tandem with an electric motor.

   - Electric Motor:

     The electric motor assists the gasoline engine, contributing extra power when needed and often rated in kilowatts (kW) or horsepower.

   - Hybrid System Output:

     This is the combined power from both the engine and the electric motor.

   - Transmission:

     Most hybrids use a continuously variable transmission (CVT) for smooth power delivery and increased efficiency.

   - Regenerative Braking:

     This system recovers energy that would otherwise be lost during braking, recharging the battery.

3. Battery Attributes:

   - Battery Type:

     - Nickel-Metal Hydride (NiMH): Found in many earlier hybrid models.

     - Lithium-Ion (Li-ion): Common in newer models for their higher energy density and lighter weight.

   - Battery Capacity:

     Measured in kilowatt-hours (kWh), capacity in non-plug-in hybrids is typically around 1.3–1.6 kWh.

   - Battery Weight:

     The battery packs can weigh roughly between 20 to 30 kilograms (or about 44–66 pounds), depending on the model.

   - Battery Voltage:

     Often in the range of 200 to 300 volts.

   - Battery Warranty and Lifespan:

     Manufacturers usually offer extended warranties (e.g., 8–10 years or 100,000–150,000 miles) due to the battery’s crucial role.

   - Battery Cooling:

     Some hybrids incorporate active (liquid) or passive cooling systems to maintain optimal battery performance.

4. Additional Attributes:

   - Driving Modes:

     Hybrids often include different modes (such as Eco, EV-only, and Power modes) that optimize performance or fuel efficiency based on driving conditions.

   - Emissions:

     Hybrids generally produce lower CO₂ emissions than conventional vehicles, with specifications provided in grams per mile or kilometer.

   - Aerodynamic Design:

     Many hybrids are designed with aerodynamics in mind, reducing drag to improve fuel efficiency.

   - Advanced Instrumentation:

     Digital displays and on-board computer systems provide real-time information about energy flow, battery charge, and fuel consumption.

**Answer for 1.12 (Gender Neutrality):**

To create a program that replaces gender-specific words in a paragraph with gender-neutral ones, I would design an algorithm that follows these steps:

1. Input and Tokenization:

   - Read the Paragraph:

     The program reads the paragraph from a text file (for example, one created in Notepad++ that is executed from the Windows command prompt).

   - Tokenize the Text:

     Split the paragraph into tokens (words) while preserving punctuation and spacing. This allows me to later reassemble the text in its original format.

2. Establish a Replacement Dictionary:

   - Prepare a dictionary (or list) that maps gender-specific words to their gender-neutral replacements. For instance:

     - “wife” → “spouse”

     - “husband” → “spouse”

     - “man” → “person”

     - “woman” → “person”

     - “son” → “child”

     - “daughter” → “child”

3. Word-by-Word Processing:

   - Check Each Token:

     For each word (token) in the text, check if it exactly matches one of the keys in the dictionary.

   - Use Exact Word Matching:

     It’s critical to match whole words only. For example, the algorithm should only replace “man” when it appears as an independent word—not as part of “woman.”

4. Perform the Replacement:

   - When a token exactly matches a gender-specific word, replace it with the corresponding gender-neutral word from the dictionary.

   - If there is no match, leave the token unchanged.

5. Reassemble the Text:

   - After processing all tokens, reassemble them back into a complete paragraph, preserving the original punctuation and spacing.

6. Output the Modified Paragraph:

   - Display or save the modified paragraph, which now contains only gender-neutral language.

How a Poorly Designed Procedure Might Generate “woperchild”:

If the algorithm does not correctly enforce whole-word boundaries, it might mistakenly perform substring replacements. For example:

- When processing the token “woman,” a naive search that replaces every instance of “man” with “person” might alter “woman” to “wo” + “person,” yielding “woperson.”

- If further replacements occur (such as replacing “son” with “child”) without proper checks, parts of different words could merge together, potentially resulting in a malformed term like “woperchild.”