

1.1 Computer Programs

- Computer program: tells a computer the sequence of steps that are needed to fulfill a task
- Hardware: the physical computer and peripheral devices
- Software: the programs that the computer executes
- Computers can store data, interact with other computers, and execute programs

1.2 Anatomy of a Computer

- Central processing unit (CPU): made of many transistors, performs program control and data processing, it locates and executes the program instructions; carries out arithmetic operations such as addition, multiplication, etc; fetches data from external memory or devices and places processed data
- Primary Storage (aka memory): made from electronic circuits that can store data, provided they are supplied with electric power
- Secondary Storage (usually a hard disk or solid-state drive): provides slower and less expensive storage that persists without electricity
- Output: computer transmits information to the user through a display screen, speakers, and printers
- Input: user can enter information for the computer using a keyboard or a pointing device such as a mouse
- Networks: through network cabling, the computer can read data and programs from central storage locations or send data to other computers
- When a program starts, its instructions are brought into its memory, where the CPU can read it. After CPU reads data and modifies it, and writes it back to memory or secondary storage

1.3 The Java Programming Language

- High level programming languages: you specify the actions that your program should carry out
- Compiler: translates the high-level instructions into the more detailed instructions (called machine code) required by CPU
- Applets: Java code that can be located anywhere on the Internet
- Java was designed so that anyone can execute programs in their browser without fear. The safety features of the Java language ensure that a program is terminated if it tries to do something unsafe
- The other benefit of Java is portability. The same Java program will run, without change, on Windows, UNIX, Linux, or Macintosh

1.4 Becoming Familiar with Your Programming Environment

Integrated development environment: the software that allows a programmer to write, edit, and test programs

Editor: a program that functions like a word processor, in which you can enter your Java instructions; you then open a console window and type commands to execute your program

```
public class HelloPrinter
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}
```

- Case sensitive: you must enter upper and lower case letters exactly as they appear in the program listing
- Running a program: Java compiler translates your source files (statements that you wrote) into class files (contains instructions for the Java virtual machine). After this, the virtual machine executes them.
- Organization: You can store programs in files. Files are stored in folders or directories.

1.5 Analyzing Your First Program

- The line “public class HelloPrinter” indicates the declaration of a class called HelloPrinter
- The word public denotes that the class is usable by the “public”
- Every source file can contain at most 1 public class, and the name of the public class must match the name of the file containing the class
- The construction “public static void main(String[] args)” declares a method called main. A method contains a collection of programming instructions that describe how to carry out a particular task
 - Every Java application must have a main method
- The main method contains one or more instructions called statements. Each statement ends in a semicolon.
- A sequence of characters enclosed in quotation marks is called a string
 - This is so that the compiler considers them plain text and does not try to interpret them as instructions
- System.out.println method prints a string or a number then starts a new line. System.out.print method prints an item without starting a new line

1.6 Errors

- Compile-time error (aka syntax errors): something is wrong according to the rules of the language and the compiler finds it
 - Ex. System.out.println(“Hello, World!”);
- Run-time error (aka logic error): program is syntactically correct but it doesn’t do what is supposed to do
 - Ex. System.out.println(“Hello, Word!”);
 - Ex. System.out.println(1/0)

1.7 Problem Solving: Algorithm Design

- An algorithm for solving a problem is a sequence of steps that is unambiguous, executable, and terminating.
- Pseudocode: informal descriptions of steps that are not yet in a language that a computer can understand
 - No strict requirements for pseudocode because it is read by human readers