CSE 06131223 ♦ CSE 06131224

Structured Programming

Lecture 2

Computer Programming Fundamentals (1)

Prepared by

Md. Mijanur Rahman, Prof. Dr.
Dept. of Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University, Bangladesh
www.mijanrahman.com

Contents

Computer Programming Fundamentals

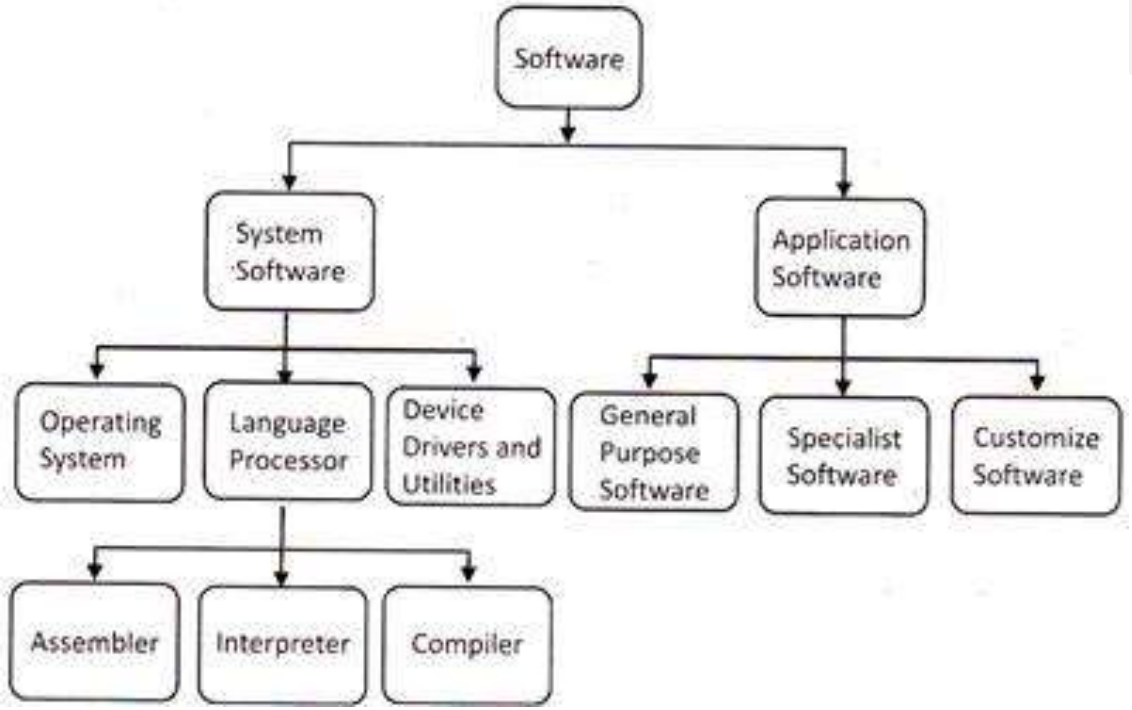
Introduction

- **Computer is an electronic device that accepts data, processes it, and generates the relevant output.** It can perform both simple and complex tasks with very high speed and accuracy.
- **The set of instructions that instruct the computer about the way the task is to be performed is called a *program***. A program is required for processing all kind of tasks—simple tasks like addition of two numbers, and complex tasks like object recognition, gaming etc.
- The instructions in a program have three essential parts:
  1. Instructions to accept the input data that needs to be processed,
  2. Instructions that will act upon the input data and process it, and
  3. Instructions to provide the output to user

Software and Programming Language

- A set of instructions that directs a computer's hardware to perform a task is called a **program, or software program**. Thus, a program is a sequence of instructions written to solve a particular problem.
- **Software** comprises the entire set of programs, procedures, and routines associated with the operation of a computer system. The two main types of software are **system software** and **application software**.
    1. **System software** controls a computer's internal functioning (through an **operating system**), and also controls such peripherals as monitors, printers, and storage devices. It also includes **programming language**.
    2. **Application software**, by contrast, directs the computer to execute commands given by the user and may be said to include any program that processes data for a user. Application software thus includes word processors, spreadsheets, database management, inventory and payroll programs, and many other applications.

Software and Programming Language



Programming Language:

- A programming language is a **computer language** that is used by **programmers (developers) to communicate with computers by writing a computer program**. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task.
- A programming language is mainly used to **develop desktop applications, websites, and mobile applications**.
- Programming language is further divided into three parts:
  1. **Low-level programming language**
  2. **High-level programming language**
  3. Middle-level programming language

Types of Programming Languages

1. Low-level Programming Language:

- Low-level language is **machine-dependent (0s and 1s)** programming language. The processor runs low- level programs directly without the need of a compiler or interpreter, so the programs written in low-level language can be run very fast. Low-level language is further divided into two parts:
  - **i. Machine Language:** Machine language is a type of low-level programming language. It is also called as **machine code or object code**. Machine language is easier to read because it is normally displayed in binary or hexadecimal form (base 16) form. It does not require a translator to convert the programs because computers directly understand the machine language programs.
  - The advantage of machine language is that it helps the programmer to execute the programs faster than the high-level programming language.
  - **ii. Assembly Language:** Assembly language (ASM) is also a type of low-level programming language that is designed for specific processors. It represents the set of instructions in a **symbolic and human-understandable form**. It uses an assembler to convert the assembly language to machine language.
  - The advantage of assembly language is that it requires less memory and less execution time to execute a program.

Types of Programming Languages

2. High-level Programming Language:

- High-level programming language (HLL) is designed for **developing user-friendly software programs and websites**. This programming language requires a compiler or interpreter to translate the program into machine language (execute the program).
  - The main advantage of a high-level language is that it is **easy to read, write, and maintain**.
  - High-level programming language includes **Python, Java, JavaScript, PHP, C#, C++, Objective C, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift programming language**.
  - A high-level language is further divided into three parts:
    i. **Procedural Oriented programming language**
    ii. **Object-Oriented Programming language**
    iii. **Natural language**

Types of Programming Languages

High-level Programming Language: Procedural Oriented programming language

- Procedural Oriented Programming (POP) language is derived from structured programming and based upon the procedure call concept. It divides a program into small procedures called **routines or functions**.
- Procedural Oriented programming language is used by a software programmer to create a program that can be accomplished by using a programming editor like IDE, Adobe Dreamweaver, or Microsoft Visual Studio.
- The advantage of POP language is that it helps programmers to easily track the program flow and code can be reused in different parts of the program.
- **Example:** C, FORTRAN, Basic, Pascal, etc.

Types of Programming Languages

High-level Programming Language: Object-Oriented Programming Language

- Object-Oriented Programming (OOP) language is **based upon the objects**. In this **programming language, programs are divided into small parts called objects**. It is used to implement real-world entities like inheritance, polymorphism, abstraction, etc in the program to makes the program resusable, efficient, and easy-to-use.
- The main advantage of object-oriented programming is that OOP is faster and easier to execute, maintain, modify, as well as debug.
- **Example:** C++, Java, Python, C#, etc.

Types of Programming Languages

High-level Programming Language: Natural Language

- Natural language is a **part of human languages** such as English, Russian, German, and Japanese. It is used by machines to understand, manipulate, and interpret human's language.
- It is used by developers to **perform tasks such as translation, automatic summarization, Named Entity Recognition (NER), relationship extraction, and topic segmentation**.
- The main advantage of natural language is that it helps users to ask questions in any subject and directly respond within seconds.

Types of Programming Languages

3. Middle-level Programming Language:

- Middle-level programming language **lies between the low-level programming language and high-level programming language**. It is also known as the intermediate programming language and pseudo-language.
- A middle-level programming language's advantages are that it supports the features of high-level programming, it is a user-friendly language, and closely related to machine language and human language.
- **Example:** C, C++ language

Commonly Used Programming Languages

Most Commonly Used Programming Languages:

- As we all know, the programming language makes our life simpler. Currently, all sectors (like education, health, financial organization, automobiles, and more) completely depend upon the programming language.
- There are dozens of programming languages used by the industries. Some most widely used programming languages are given below:

1. C Language:

- C is a **popular, simple, and flexible general-purpose computer programming language. Dennis M Ritchie develops it in 1972** at AT&T. It is a combination of both low-level programming language as well as a high-level programming language. It is used to design applications like **Text Editors, Compilers, Network devices, and many more**.

Commonly Used Programming Languages

Most Commonly Used Programming Languages:

2. C++ Language:

- C++ is one of the thousands of programming languages that we use to develop software. C++ programming language is developed by **Bjarne Stroustrup in 1980**. It is similar to the C programming language but also includes some additional features such as **exception handling, object-oriented programming, type checking, etc**.

3. Java Language:

- Java is a simple, secure, platform-independent, reliable, architecture-neutral high-level programming language **developed by Sun Microsystems in 1995**. Now, Java is owned by Oracle. It is mainly used to develop bank, retail, information technology, android, big data, research community, web, and desktop applications.

Commonly Used Programming Languages

Most Commonly Used Programming Languages:

4. C# Language:

- C# (**pronounced as C sharp**) is a modern, general-purpose, and object-oriented programming language used with XML based Web services on the .NET platform. It is mainly designed to improve productivity in web applications. It is easier to learn for those users who have sufficient knowledge of common programming languages like C, C++, or Java.

5. JavaScript Language:

- **JavaScript** is a type of **scripting language** that is used on both client-side as well as a server-side. It is developed in the **1990s** for the Netscape Navigator web browser. It allows programmers to implement complex features to make web pages alive. It helps programmers to create dynamic websites, servers, mobile applications, animated graphics, games, and more.

Commonly Used Programming Languages

Most Commonly Used Programming Languages:

6. HTML Language:

- **HTML stands for Hyper Text Markup Language**. HTML is the standard markup language for creating Web pages. It describes the structure of a Web page and its contents. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

7. PHP Language:

- PHP stands for **Hypertext Preprocessor**. It is an open-source, powerful server-side scripting language mainly used to create static as well as dynamic websites. It is developed by **Rasmus Laird in 1994**. Inside the php, we can also write HTML, CSS, and JavaScript code. To save php file, file extension .php is used.

Commonly Used Programming Languages

Most Commonly Used Programming Languages:

8. Python Language:

- Python is one of the most widely used user-friendly programming languages. It is an open-source and easy to learn programming language developed in the 1990s. It is **mostly used in Machine learning, Artificial intelligence, Big Data, GUI based desktop applications, and Robotics**.

9. R Language:

- R is one of the popular programming languages that is used in **data analytics, scientific research, machine learning algorithms, and statistical computing.** It is developed in 1993 by Ross Ihaka and Robert Gentleman. It helps marketers and data scientists to easily analyze, present, and visualize data.

Commonly Used Programming Languages

Most Commonly Used Programming Languages:

10. Ruby Language:

- Ruby is an open-source, general-purpose, and pure object-oriented programming language released in **1993**. It is used in front-end and back-end web development. It is mainly **designed to write CGI (Common Gateway Interface) scripts**.

11. Go Language:

- Go or Golang is an **open-source programming language**. It is used to build simple, reliable, and efficient software. It is developed by **Robert Griesemer, Rob Pike, and Ken Thompson in 2007**.

Program Development Life Cycle

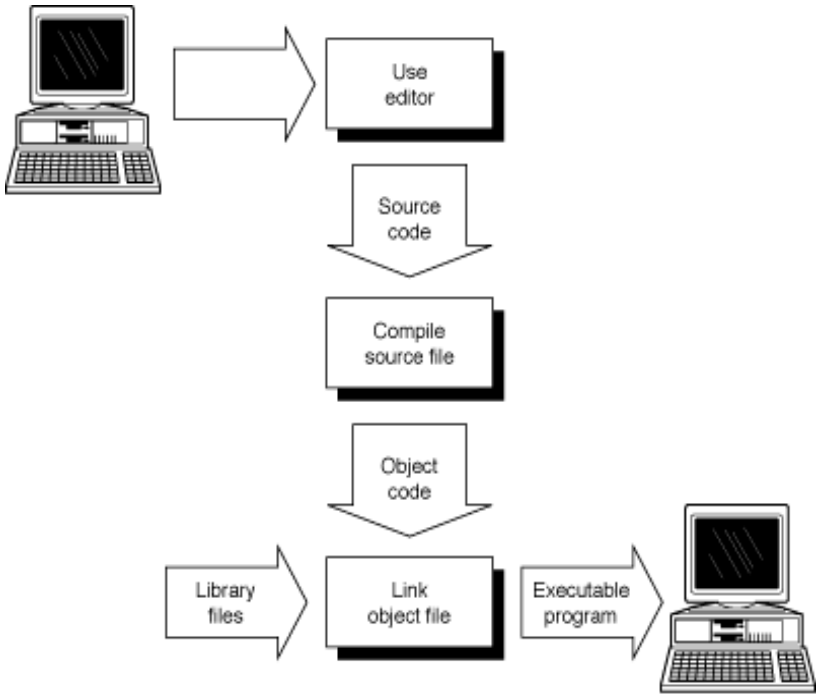


Fig. *The C source code that you write is converted to object code by the compiler and then to an executable file by the linker.*

- **The Program Development Cycle has its own steps.**
  1. In the first step, you use an editor to create a disk file containing your source code.
  2. In the second step, you compile the source code to create an object file.
  3. In the third step, you link the compiled code to create an executable file.
  4. The fourth step is to run the program to see whether it works as originally planned.

Program Development Life Cycle

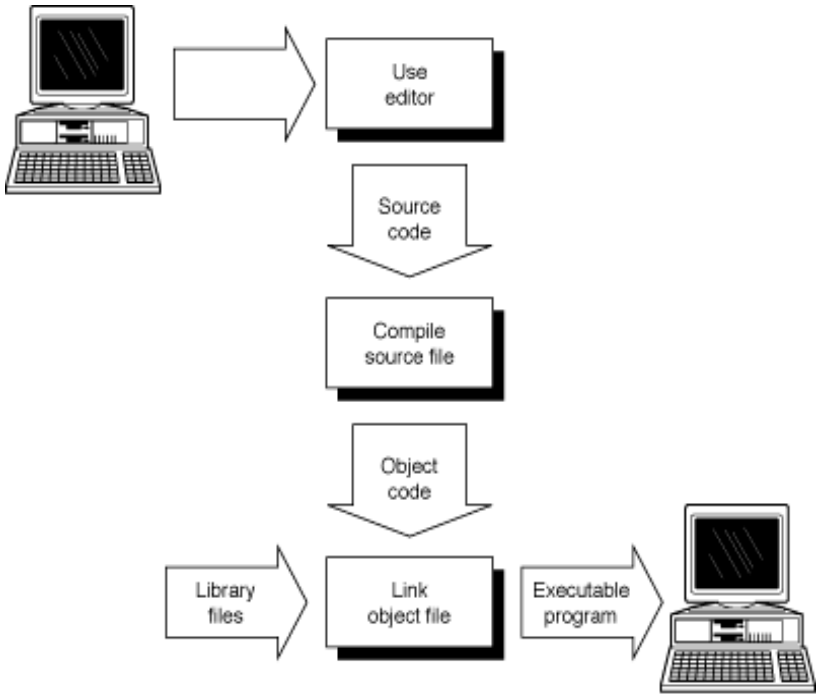


Fig. *The C source code that you write is converted to object code by the compiler and then to an executable file by the linker.*

1. Creating the Source Code

- *Source code* is a series of statements or commands that are used to instruct the computer to perform your desired tasks. The first step in the Program Development Cycle is to enter source code into an editor.

2. Compiling the Source Code

- A computer requires digital, or *binary,* instructions in what is called ***machine language***. Before run your program, it must be translated from source code to machine language. This translation is performed by a program called a ***compiler***.
- The compiler takes your source code file as input and produces a disk file containing the machine language instructions that correspond to your source code statements. **The machine language instructions created by the compiler are called *object code,* and the disk file containing them is called an *object file* (.OBJ extension).**

Program Development Life Cycle

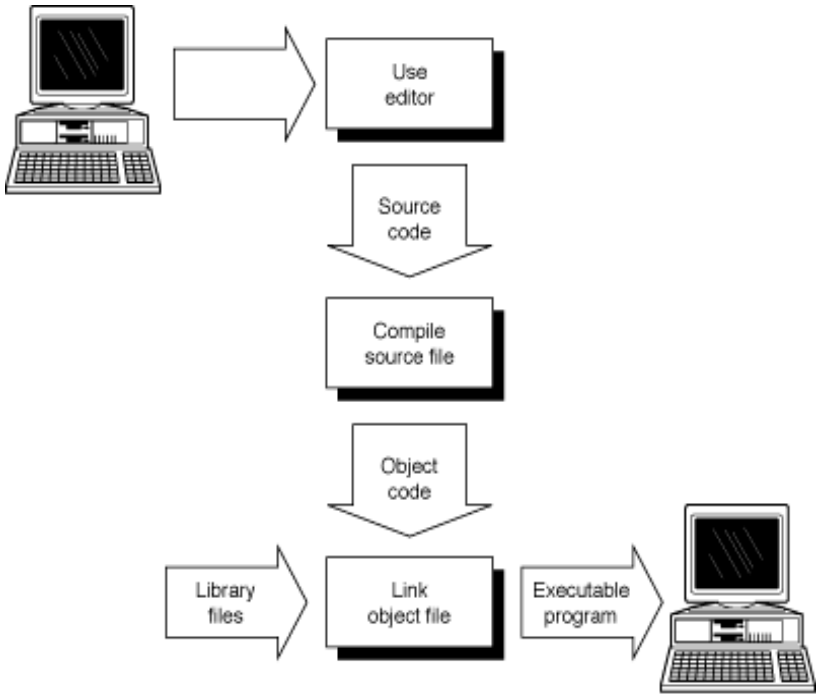


Fig. *The C source code that you write is converted to object code by the compiler and then to an executable file by the linker.*

3. Linking to Create an Executable File

- A part of the C language is a function library that contains *object code* for predefined functions. If your program uses any of these functions, the object file produced when your source code was compiled must be combined with object code from the function library to create the final executable program. This process is called *linking,* and it's performed by a program called a *linker*.

4. Executing/Run the program

- Once your program is compiled and linked to create an executable file, you can run it.
- If you run the program and receive results different from what you thought you would, you need to go back to the first step. You must identify what caused the problem and correct it in the source code. You keep following this cycle until you get the program to execute exactly as you intended.

Types of Errors in Programs

- **Syntax Errors:** Violation of syntactic rules in a Programming Language generates syntax errors. Compiler helps user to identify the Syntax error and correct it. A syntax error is an error in the typing of the code or statement.
- For example: In C, "prinff()“ would be a syntax error, done by mistyping. It would be “printf()”.

- **Commonly occurred syntax errors are:**
  - If we miss the parenthesis (}) while writing the code.
  - Displaying the value of a variable without its declaration.
  - If we miss the semicolon (;) at the end of the statement.

Types of Errors in Programs

- **Semantic Errors:** Semantic errors are the errors that occurred when the statements are not understandable by the compiler. Doing logical mistakes causes semantic errors. The Compiler cannot notice these errors, but on execution, they cause unexpected wrong results. These errors can only be corrected by the careful programmer.
- A semantic error basically means invalid logic. For example: 2+2=1 would be a semantic error because it is incorrect logic.
- **Run-time Errors:** Sometimes the errors exist during the execution-time even after the successful compilation known as run-time errors. When the program is running, and it is not able to perform the operation is the main cause of the run-time error.
- The division by zero is the common example of the run-time error. These errors are very difficult to find, as the compiler does not point to these errors.

THE END