



BASICS OF C LANGUAGE

Phases in the development of a program, Compilers and Interpreters

Phases in the Program Development

- **Program development life cycle** is a systematic way of developing **quality programs**. It provides an organized plan for breaking down the task of program development into **manageable parts**, each of which must be successfully completed before moving on the next phase.

Phases in the Program Development

- The **program development process** is divided into following phases:-
 - a. Defining the problem
 - b. Designing the program
 - c. Building the program
 - d. Testing the program
 - e. Documenting the program
 - f. Deploying and maintaining the program

Phases in the Program Development

- **Defining the problem**

- The first step in developing a program is to define the problem. In major software projects, **this is a job for system analyst**, who provides the results of their work to programmers in the form of a program specification.
- The **program specification** precisely defines the input data, the processing that should take place, the format of the output reports and the user interface.

Phases in the Program Development

- **Defining the problem (...continued)**
 - Depending on the size of the job, the program development might be handled by an individual or a team.

Phases in the Program Development

- **Designing the Program**
 - Program design **begins** by focusing on the main goal that the program is trying to achieve and then breaking the program into manageable components, each of which **contributes to this goal**.
 - This approach of program design is called *top-down program design or modular programming*.

Phases in the Program Development

- **Designing the Program**

- For each module, **the programmer draws a conceptual plan** using an appropriate program design tool to visualize how the module will do its assigned job. The various familiar program design tools are:

- *Structure Charts*
- *Algorithms*
- *Flowcharts*
- *Pseudo-codes*

Phases in the Program Development

- **Building the Program**

- Once the design of the program is ready, **the next step is to** convert the program design into a computer program.
- **During this conversion** each of the steps of the program design is coded as one or more C language instructions.

Phases in the Program Development

- **Building the Program**
 - It includes:
 - Creating and editing program
 - Compiling program
 - Linking program
 - Executing program

Phases in the Program Development

- Building the Program

- It includes:

- Creating and editing program

- Once the program is ready on the paper, we key in computer memory using a Text Editor. A Text Editor helps us to enter the character data into computer memory, allows editing the data in computer memory, and save the data from memory in a disk file on secondary memory with extension “.c”.

- *This stored file is known as source file, and its contents are known as source code.*

Phases in the Program Development

- **Building the Program**

- It includes:

- **Compiling program**

- The source code in the source file, stored on the disk, must to be translated into machine language. This job is done by the **compiler**. The C compiler actually is a combination of two separate programs – the **preprocessor** and the **translator**.

- » The preprocessor reads the source code and prepares it for translation.

- » The translator reads the translation unit instruction-by-instruction and checks them for their grammatical accuracy.

Phases in the Program Development

- **Building the Program**

- It includes:

- **Compiling program**

- Therefore, if there is even a single syntax error, the translation process, known as compilation, is terminated.

- » However, if there are no syntax errors in the translation unit, the translator rereads the instruction from the beginning, translates them into machine language, and writes them onto a disk file.

- » The translated version of the source code is known as object code, and is stored in the disk file with extension “.obj”.

Phases in the Program Development

- **Building the Program**

- It includes:

- **Linking program**

- Once the **source code is translated into object code**, though it is in machine language, still it is not in executable form.
- The reason being is that it may be referring to other functions which are not included in the **object code**. These **can be library functions** or **user-defined functions**. All these functions also need to be included in the object code to get a final machine code which is in the executable form, known as **executable code**. **This task is performed by linker.**

Phases in the Program Development

- **Building the Program**

- It includes:

- Linking program

- A **linker** is a **development tool** that **extracts** the referred **library functions** from the **system libraries**, **reads** the **object codes** of the **user-defined functions** from the **object files** and **integrates** them with the current object code to produce the final executable code, which is stored in disk file with extension **“.exe”**.

- » This ***executable code is the final form of the program that is ready for execution.***

Phases in the Program Development

- **Building the Program**

- It includes:

- Executing program

- Once the program is linked, **it is ready for execution**. To execute a program we give an operating system command, such as **run**, to load the program into computer memory and execute it.
- **Getting the program into memory is the function of an operating system program known as loader**. The loader locates the executable program in the secondary storage, reads it and brings into computer memory.
- Once the program is loaded, the **operating system transfers the control** to the program and the program begins its execution.

Phases in the Program Development

- **Testing the Program**

- Even **when the program is executing**, the **output of the program may not be correct**. This will be because of **logical errors** in the program.

- A **logical error** is a mistake that the programmer made while designing the solution to the problem.

- Therefore, the **programmer must find and correct logical errors by carefully examining the program output for a set of data for which results are already known**.

Phases in the Program Development

- Documenting the Program
 - After testing, the program development is almost complete. The structure charts, algorithms, pseudo-codes and flowcharts developed during the design phase become documentation for others who are associated with the program.

Phases in the Program Development

- Documenting the Program
 - This phase ends by writing a manual that provides an overview of the program's functionality, tutorials for the beginner, in-depth explanations of major program features, reference documentation of all program commands and a thorough description of the error messages generated by the program.
 - These forms of documentation are known as external documentation.

Phases in the Program Development

- Documenting the Program
 - In addition to external documentation, a program may be documented internally.
 - Internal documentation of a program includes the comments written in the program code.

Phases in the Program Development

- **Deploying and Maintaining the Program**
 - In the **final phase**, the **program is deployed (installed)** at the user's site.
 - Here also, the program is kept under watch till the user gives **green** signal to it.
 - Users may **discover errors that were not caught in the testing phase**, no matter how exhaustively the program was tested.

Phases in the Program Development

- **Deploying and Maintaining the Program**
 - Even after the **project development is complete**, it needs **to be** maintained **and** evaluated regularly.
 - In program maintenance, the **programming team** **fixes program errors** that users discovers during its day-to-day use.

Practice Questions

- Where was C originally developed and by whom?
- What is meant by batch processing and interactive computing?
- Contrast **Machine Level**, **High Level** and **Assembly Level** Languages. Which of these languages are hardware dependent ?
- Define source code and object code. Which type of code will be preferred by you and your computer?
- **Contrast Interpreter, Compiler and Assembler.**