

Programming Fundamentals

Lecture # 1

Course Information

- Title of Course : Programming Fundamentals
- Course Code : CS111
- Credit Hours : 4
- Paradigm Studied: Structure Oriented
- Language Used : C++

Marking Criteria

- Assignments 10 Points
- Quizzes(Un announced) 10 Points
- Mid-Term exam 20 Points
- Final-Term exam 60 Points

Books

- how to program in c++ by dietel and dietel(3rd OR latest Edition)
- C++ Programming from Problem Analysis to Program Design by D S Malik(5th Edition)

Course Content

- Data Types
- Conditional Statements
- Repetition Structures
- Functions
- Arrays
- String Manipulation
- Structures
- Pointers
- File Handling

What is Computer?

- An electronic device which is capable of receiving data in a particular form and of performing a sequence of operations in accordance with a predetermined but variable set of procedural instructions (program) to produce a result in the form of information.

Computer Organization

- Input Unit
- Output Unit
- Memory Unit
- Arithmetic and logical unit(ALU)
- Central processing unit(CPU)

Computer Programming Languages

- Two types of programming languages:
- Low-level language
- High-level language

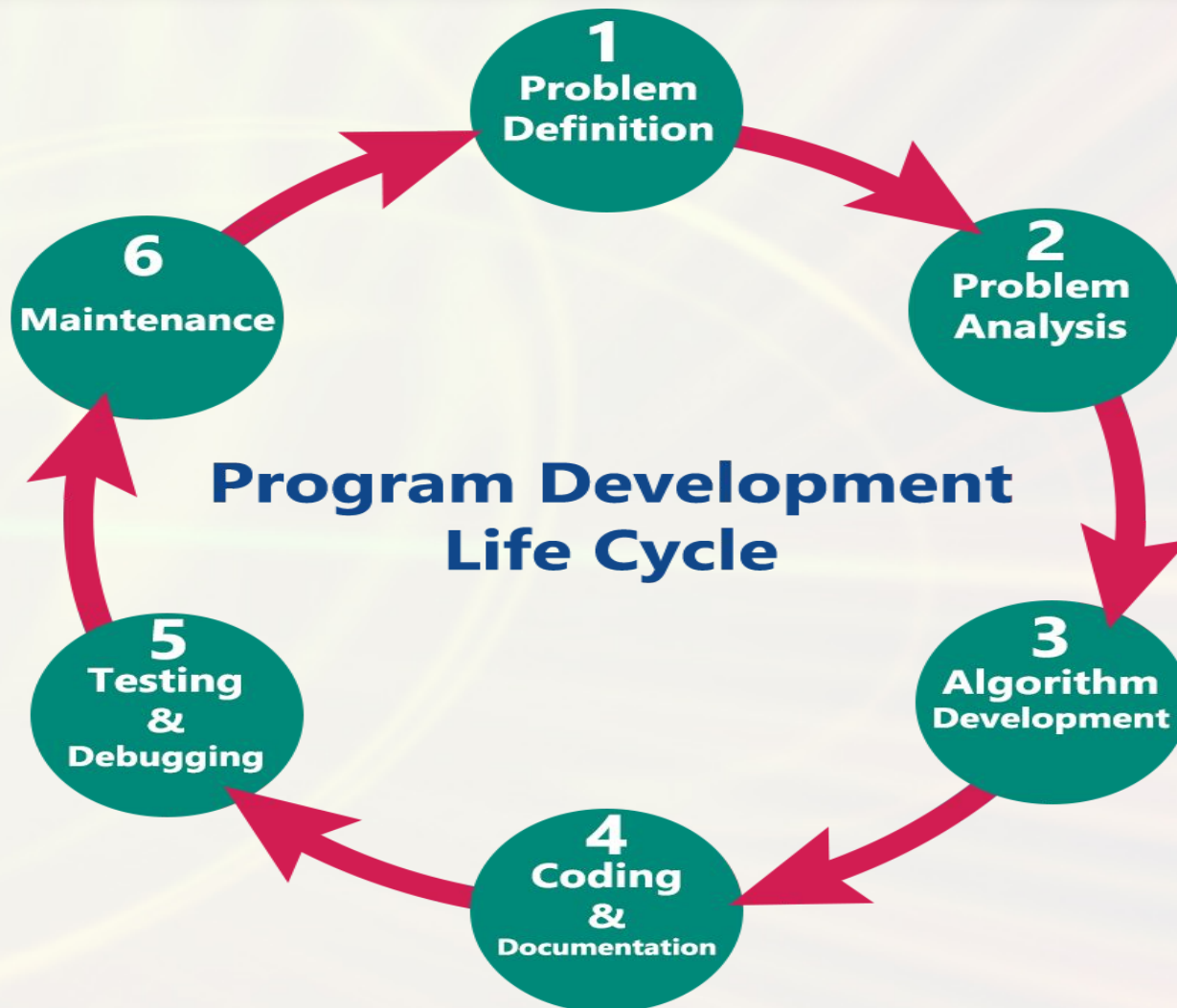
Low level languages

- These languages are near to computer hardware and far from humans
- Computer can understand these languages easily.
- E.g.
 - Assembly language

High level languages

- Those languages which are near to humans are called high level languages.
- Instruction in these languages are written in English like words
- High level languages are further divide into following languages,
 - Procedural languages
 - Object-Oriented languages
 - Non-Procedural languages

Program Development Life Cycle



What is Paradigm?

- A programming **paradigm** is a style, or “way,” of programming.
- Never use the phrase “programming language **paradigm**.”
- A **paradigm** is a way of doing something (like programming), not a concrete thing (like a language).

What is Programming Language?

- A **programming language** is a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks.
- The term **programming language** usually refers to high-level languages, such as BASIC, C, C++, C#, Java, FORTRAN, Ada, and Pascal.
- In this course we will study C++ programming language.

IDE

- Integrated development environment
- Visual studio(Latest 2017 community edition)

Background History of C++

- C++ Development started in 1979.
- During the creation of Ph.D. thesis, Bjarne Stroustrup worked with language called Simula.
- Simula is basically useful for the simulation work.
- Simula was first language to support object-oriented programming paradigm
- Bjarne Stroustrup identified that this OOP features can be included in the software development.
- After that Bjarne Stroustrup started working on the C language and added more extra OOP features to the classic C.
- He added features in such a fashion that the basic flavour of C remains unaffected.
- C++ includes some add-on features such as classes, basic inheritance, in-lining, default function arguments, and strong type checking

Basic History of C++

- During 1970 Dennis Ritchie created C Programming language.
- In the early 1980's, also at Bell Laboratories, another programming language was created which was based upon the C language.
- C++ is also called as C with classes
- Stroustrup states that the purpose of C++ is to make writing good programs easier and more pleasant for the individual programmer.
- C++ programming language is extension to C Language.
- In C we have already used increment operator (++). Therefore we called C++ as “Incremented C” means Extension to C.

Types of Code

- There are two types of code:
- Source code
 - A program written in a high-level language is called source code. Source code is also called source program. Computer cannot understand the statements of high-level language.
 - The source code cannot be executed by computer directly. It is converted into object code and then executed.
- Object code
 - A program in machine language is called object code. It is also called object program or machine code. Computer understands object code directly.

Language Processor

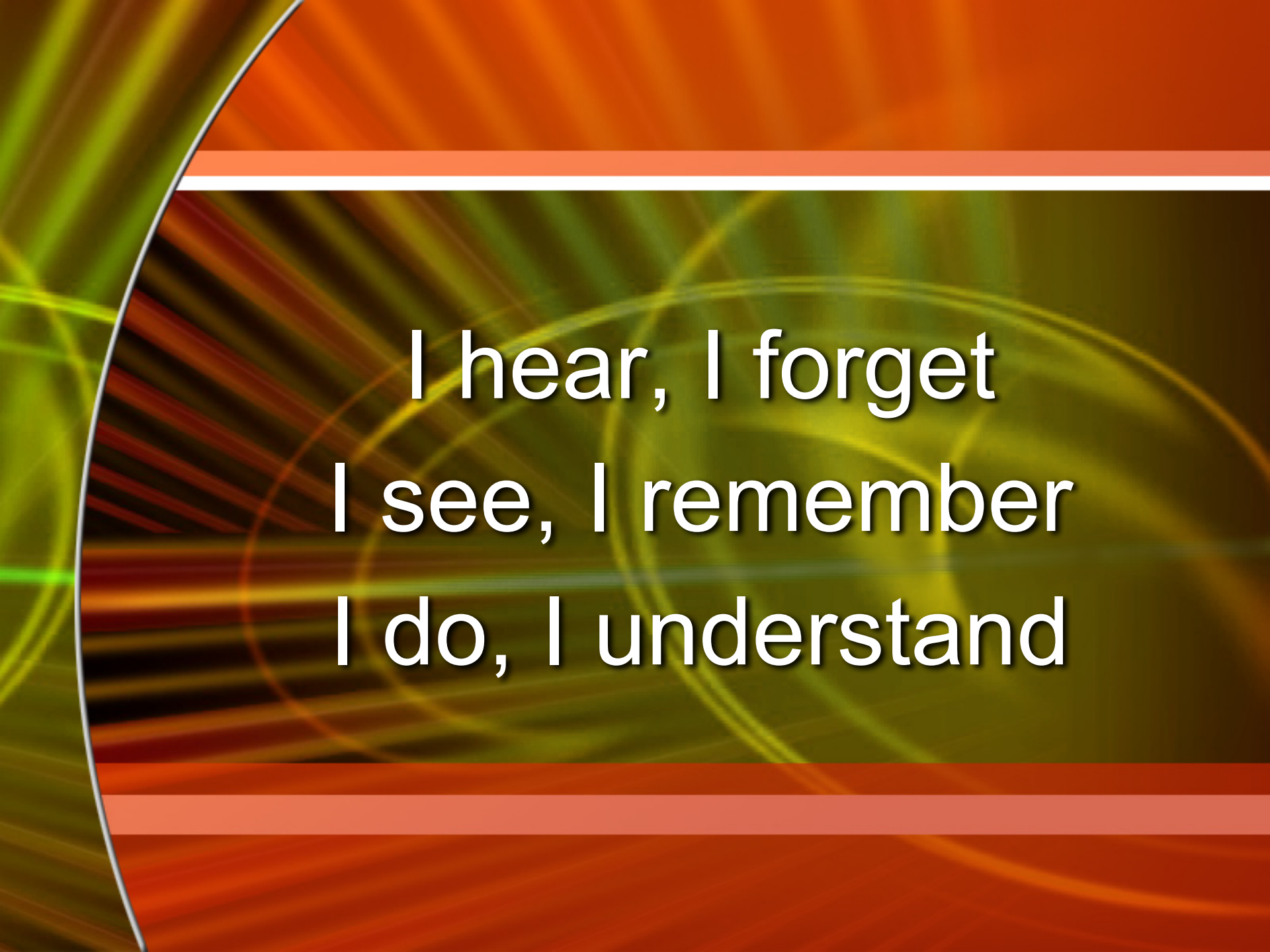
- Computer understands only machine language.
- A program written in high-level cannot be run on a computer directly. It must be converted into machine language before execution.
- Language processor or translator is a software that converts these programs into machine language.
- Every computer language has its own translators.
- Different types of language processors are as follows:
 - Compiler
 - Interpreter

Compiler

- A compiler is a program that converts the instruction of a high level language into machine language as a whole.
- A program written in high-level language is called source program.
- Compiler converts source program into machine code known as object program.
- The compiler checks each statement in the source program and generates machine instructions.
- Compiler also checks syntax errors in the program.
- A source program containing an error cannot be compiled.

Interpreter

- An interpreter is a program that converts one statement of a program at one time. It executes this statement before translating the next statement of the source program.
- If there is an error in the statements, the interpreter stops working and displays an errors message.
- The advantage of interpreters over compilers is that an error is found immediately. So the programmer can correct errors during program development.
- The disadvantage of interpreter is that it is not very efficient. The interpreter does not produce an object program. It must convert the program each time it is executed.



I hear, I forget
I see, I remember
I do, I understand