

IT1010 – Introduction to Programming

Lecture 01 – Note

COMPUTERS AND PROGRAMS

A computer is a machine that can be programmed to perform tasks. A program is a set of instructions that tells a computer what to do. The relationship between computers and programs is that programs allow computers to perform tasks. Without programs, computers would be useless.

Here is a simple analogy to help you understand the relationship between computers and programs:

- A computer is like a car.
- A program is like a driver's manual.

Just like a car needs a driver's manual to tell it how to drive, a computer needs a program to tell it what to do. Without a driver's manual, a car would be useless. Without a program, a computer would be useless.

Here are some examples of programs:

- The operating system is a program that controls the basic functions of a computer.
- A word processor is a program that allows you to create and edit documents.
- A web browser is a program that allows you to view web pages.
- A game is a program that allows you to play games.

As you can see, programs allow computers to do a wide variety of things. Without programs, computers would be very limited in their capabilities.

PROGRAMMING AND PROGRAMMING LANGUAGES

Programming is the process of creating a set of instructions that tells a computer what to do. These instructions are written in a special language called **a programming language**.

Programming languages are like human languages, but they are much more precise. Every word and symbol in a programming language has a specific meaning. This allows computers to understand the instructions and carry them out exactly.

Programming can be used to create a wide variety of things, including:

- Software applications

- Web pages
- Games
- Operating systems
- Scientific simulations

Here is a simple analogy to help you understand programming:

- A programming language is like a foreign language.
- A computer is like a foreigner.
- A program is like a message that you want to send to the foreigner.

Just like you need to learn a foreign language in order to communicate with a foreigner, you need to learn a programming language in order to program a computer. And just like a message needs to be written in a language that the foreigner understands, a program needs to be written in a programming language that the computer understands.

HOW TO PROGRAM

Here are the basic steps on how to write programs:

1. Understand the problem you are trying to solve. What do you want the program to do? What inputs will it need? What outputs will it produce?
2. Design a solution. How will you solve the problem? What steps will the program need to take?
3. Write code. Now that you have a clear understanding of your solution, you can start writing code in a programming language.
4. Test and debug. Once you have written some code, you need to test it to make sure that it works correctly. You will also need to debug your code to find and fix any errors.
5. Repeat steps 4-5 until the program is working correctly.

Here are some additional tips for writing programs:

- Use a programming language that is appropriate for the task at hand.
- Keep your code organized and easy to read.
- Use comments to explain what your code is doing.
- Use a debugger to help you find and fix errors.
- Test your code thoroughly.

SOFTWARE DEVELOPMENT LIFE CYCLE

Here are the six steps of the program development life cycle (SDLC) :

1. Analysis

The analysis phase is where you gather requirements and understand the problem you are trying to solve. You will need to talk to stakeholders, gather documentation, and perform other research to understand the problem domain.

2. Design

The design phase is where you create a blueprint for the program. This includes defining the architecture, designing the user interface, and designing the database.

3. Implementation

The implementation phase is where you write the code for the program. This is where you put your design into action.

4. Testing

The testing phase is where you test the program to make sure it works correctly. This includes unit testing, integration testing, and system testing.

5. Release

The release phase is where you deploy the program to production. This is where the program is made available to users.

6. Maintenance

The maintenance phase is where you update and maintain the program. This includes bug fixing, adding new features, and making other changes.

ALGORITHMS

An algorithm is a step-by-step procedure that tells a computer what to do. Algorithms are used in all sorts of programs, from simple calculators to complex search engines.

Algorithms can be written in a variety of ways, including natural language, pseudocode, and programming languages. Natural language algorithms are written in plain English, while pseudocode algorithms are written in a special notation that is similar to programming languages. Programming languages are the most common way to write algorithms, as they can be executed by computers.

Algorithms are an essential part of computer science. They are used to solve a wide variety of problems, from simple to complex. If you are interested in learning more about algorithms, there are many resources available online and in libraries.

Here are some additional characteristics of algorithms:

- They are precise. The steps in an algorithm must be clear and unambiguous.
- They are finite. An algorithm must terminate after a finite number of steps.

- They are general. An algorithm should be able to solve a problem for any set of input data.
- They are efficient. An algorithm should be as fast and as accurate as possible.

Here is an example of an algorithm in natural language to make tea:

Code snippet

To make tea, you will need:

- * Tea leaves
- * Water
- * A kettle
- * A mug
- * A spoon

Here are the steps:

1. Gather your materials.
2. Boil water in the kettle.
3. While the water is boiling, place the tea leaves in the mug.
4. Pour the boiling water over the tea leaves.
5. Let the tea steep for 3-5 minutes.
6. Remove the tea leaves from the mug.
7. Add milk and sugar to taste.
8. Enjoy your tea!