



## COMPUTER PROGRAMMING

### Chapter I: Introduction to Computer Programming

#### 1. Definition of Computer Programming:

Computer programming is the process of writing instructions that a computer can execute in order to perform specific tasks. These instructions are written using programming languages such as:

- Python
- C
- Java
- JavaScript

A program is a set of instructions that tells the computer how to solve a problem or perform an action.

Examples of tasks performed by programs:

- Calculating numbers
- Managing databases
- Running websites
- Controlling devices

Programming is essential in modern technology including:

- mobile applications
- operating systems
- artificial intelligence
- video games

#### 2. History of Programming:

Programming evolved with the development of computers.

##### Important milestones:

Year	Event
1940s	First electronic computers
1957	FORTRAN language created
1972	C programming language created
1991	Python language released
2000s	Growth of web programming
Today	AI, data science, cloud computing

Early programming required machine language, which was difficult for humans to read. Modern languages are higher-level, meaning they are easier for humans to understand.

#### 3. Types of Programming Languages:

Programming languages can be classified into different categories.

##### Low-Level Languages:

Closer to machine hardware.

Examples: Assembly language

Advantages:

- Very fast
- Efficient

Disadvantages:

- Difficult to write
- Hard to maintain

##### High-Level Languages:

Closer to human language.

Examples:

- Python
- Java
- C++
- JavaScript

Advantages:

- Easy to read
- Faster development
- Portable across systems

#### 4. Compiled vs Interpreted Languages:

Programming languages execute code in different ways.

Type	Description	Example
Compiled	Code is converted into machine code before execution	C, C++
Interpreted	Code runs line by line using an interpreter	Python

Compiled languages are usually faster, while interpreted languages are easier to debug.

## Chapter II: Basic Programming Concepts

### 1. Algorithms:

An algorithm is a step-by-step procedure used to solve a problem.

Example algorithm for adding two numbers:

1. Start
2. Input number A
3. Input number B
4. Compute A + B
5. Display result
6. End

[ Start ] → [ Input A ] → [ Input B ] → [ A + B ] → [ Display Result ] → [ End ]

Algorithms help programmers plan solutions before coding.

### 2. Variables and Data Types:

A variable is a container used to store data.

Example in Python:

```
age = 20          name = "Sara"
```

Common data types:

Data Type	Example
Integer	10
Float	3.14
String	"Hello"
Boolean	True / False

### 3. Operators:

Operators perform operations on variables. Examples:

Operator	Meaning	Example
+	Addition	5 + 2
-	Subtraction	7 - 3
*	Multiplication	4 * 2
/	Division	10 / 2

Comparison operators:

- = equal
- > greater than
- < less than

### 4. Control Structures:

Control structures determine the flow of a program.

#### Conditional Statements:

Used for decision making.

Example in Python:

```
if age >= 18: print ("Adult")
else: print ("Minor")
```

#### Loops:

Loops repeat instructions.

Example: for i in range (5): print (i)

Types of loops:

- for loop
- while loop

## Chapter III: Functions and Modular Programming

### 1. Functions:

A function is a reusable block of code designed to perform a specific task.

Example in Python:

```
def greet(name):    print ("Hello", name)
```

Calling the function: greet("Ali")

Benefits of functions:

- code reuse
- easier debugging
- better organization

## 2. Parameters and Return Values:

Functions can receive inputs (parameters) and return results.

Example: def add (a, b): return a + b

Usage: result = add (5,3)

## 3. Modular Programming:

Modular programming divides a program into separate modules or functions.

Advantages:

- easier maintenance
- code reuse
- better readability

Large programs are often composed of many modules working together.

## Chapter IV: Introduction to Python Programming

### 1. Why Python?

Python is one of the most popular programming languages.

Reasons:

- simple syntax
- easy to learn
- powerful libraries
- widely used in AI and data science

Python is used in:

- web development
- machine learning
- automation
- scientific computing

### 2. Basic Python Program:

Example: Print ("Hello World")

This program prints text on the screen.

### 3. User Input Example:

```
name = input ("Enter your name: ") print ("Hello", name)
```

The program asks the user for input and prints a message.

## Chapter V: Applications of Programming

Programming is used in many fields.

Field	Description	Examples / Uses
<b>Software Development</b>	Programming used to create computer applications and system software.	Word processors, media players, operating systems
<b>Web Development</b>	Programming used to build and maintain websites and web applications.	HTML, CSS, JavaScript, Python
<b>Artificial Intelligence</b>	Programming used to create intelligent systems that can learn and make decisions.	Machine learning, data analysis, robotics
<b>Game Development</b>	Programming used to design and build video games and interactive entertainment.	Graphics, physics, interaction

**Example:**

```

Python
1 # bot.py
2
3 from chatterbot import ChatBot
4
5 chatbot = ChatBot("Chatpot")
6
7 exit_conditions = (":q", "quit", "exit")
8 while True:
9     query = input("> ")
10    if query in exit_conditions:
11        break
12    else:
13        print(f"👉 {chatbot.get_response(query)}")

```

## **Conclusion:**

Computer programming is a fundamental skill in modern technology.

It allows developers to create applications, automate tasks, and solve complex problems.

Key concepts include:

- algorithms
- variables and data types
- control structures
- functions
- programming languages such as Python

Understanding these foundations helps students build more advanced knowledge in software engineering, artificial intelligence, and data science.