Tishk International University
IT Department
Course Code: IT-117

Programming I

Lecture 1

Introduction To Programming

Fall 2023

Hemin Ibrahim
Hemin.Ibrahim@tiu.edu.iq

About Me

Hemin Ibrahim

- BSc in Mathematics
- BSc in Information Technology
- MSc(Eng) in Advanced Software Engineering
- PhD in Artificial Intelligence

About Me

Hemin Ibrahim

- hemin.ibrahim@tiu.edu.iq
- (WhatsApp)
  - Sending text messages **ONLY**.
  - Phone calls are **NOT** an option.
  - Voice messages are **NOT** permitted.
- Office: 252

Course policy - Mobile

The following actions are NOT allowed.

- Answering call
- Sending messages
- Watching Youtube
- Taking a photo (only with permission)

Your Cellphone should be silent

Course policy - Attendance

- **I have a big problem with attendance**
  - Ahmmm
- **I will fail if I don’t have 70%**
  - Ahmmm
- **Can you please help me**
  - **NO**

There will be no attendance record if you are unavailable

Course policy - Class time

Don’t be Late

Be on time for the lecture

Remember, if class starts at 10:00AM, 10:06, means you are late

AND

10:10 You Consider as absent

Course policy - Lecture

- **Raise your hand if you need to step out of the class.**
- **Ensure that you arrive on time for the second part of the lecture**

Outline

- What is Programming
- Why Learn Programming
- Types of Programming Languages
- Advantages of learning C++ language
- IDE (Integrated Development Environment)
- Introduction to C++
- Comments
- First C++ example code
- Special Characters

Objectives

- Provide an introduction to the concept of programming.
- Emphasize the importance and relevance of learning programming.
- Explore the various types of programming languages.
- Highlight the advantages of the C++ programming language.
- Explore into the basics of C++, covering comments and special characters.
- Foster a foundational understanding of programming concepts, with a specific focus on the practical applications of C++.

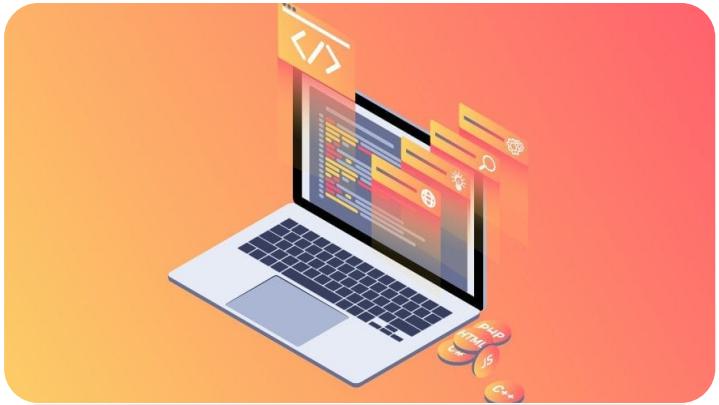
What is programming?

Programming is the process of giving instructions to a computer to perform specific tasks.

- Instructions must be written in a way the computer can understand
- Programming languages are used to write programs

Why Learn Programming?

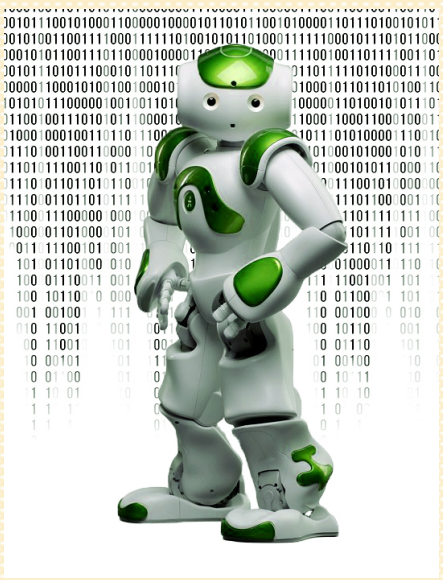
- Problem Solving
- Creativity
- Versatility
- Personal Projects

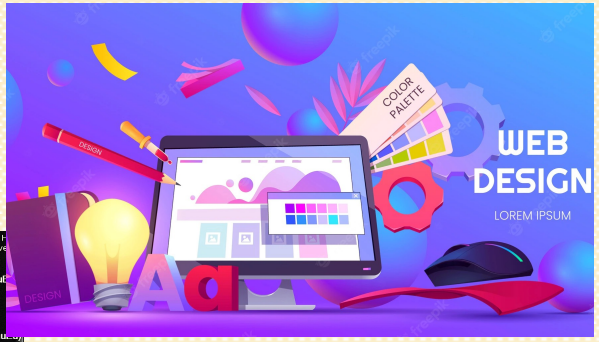


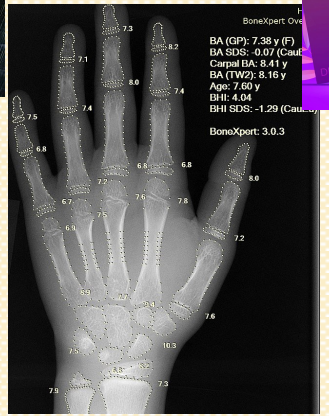
Why Is Programming Important?

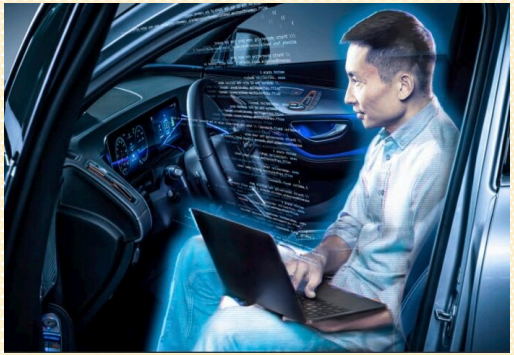
physics-I
Calculus-I
Programming-I
Computer-Skills
Intro to IT II
Calculus II
General Physics II
Kurdology II
Turkish II
Programming Fundamentals
Technical English
Computer Hardware
Logic Design
Programming-I
Database-I
Multimedia Technologies
Discrete Math
Academic Writing Skills I
Programming II
Web Design
Database System II
Probability and Statistics
Academic Writing Skills II

Computer Aided Design
Object Oriented Programming-I
Data Structure & Algorithm-I
Operating System
E-Business/E-Commerce
Data Communication & Networking-I
Data Structures & Algorithms II
Object Oriented Programming II
Data Communication & Networking II
Web Programming
Human Computer Interaction
Research Methodology
IT Project Management
Introduction to AI & Robotics
Open Source
Web Technologies
Cloud Computing
Wireless Networking
Information Security
Software Engineering
Cloud Computing




COLOR PALETTE
WEB DESIGN
LOREM IPSUM


BA (GP): 7.38 y (F)
BA SDS: -0.07 (Caut
Carpal BA: 8.41 y
BA (TW2): 8.16 y
Age: 7.60 y
BHI: 4.04
BHI SDS: -1.29 (Cau
BoneXpert: 3.0.3




CALL OF DUTY
MOBILE


SOFTWARE
UX
XML
CSS
C++
<code/>
Ai
Ps
UI

Why Is Software Important?

- Most electronic devices run software (vs Hardware)
- Software bugs easy to fix (vs Hardware)
- Allows new features to be added later (vs Hardware)
- More software jobs (vs Hardware)
- A small team of hardware can design a product like iPhone, BUT thousands of individuals and companies develop apps

Types of Programming Languages

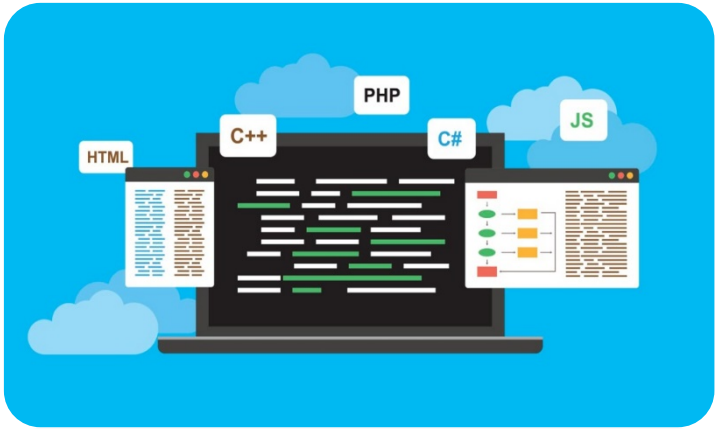
- Programming languages come in various types, each serving specific purposes and catering to different needs.

1) Low-level Programming Languages

- Machine Learning
- Assembly Language

2) High-level Programming Languages

- Are closer to human languages
- C, C++, Java, Dart …



Types of Programming Languages (Cont.)

3) Compiled vs Interpreted languages

- Compiled languages: Code written in compiled languages is translated into machine code before execution.

  Ex) C, Java, C++, and Rust.

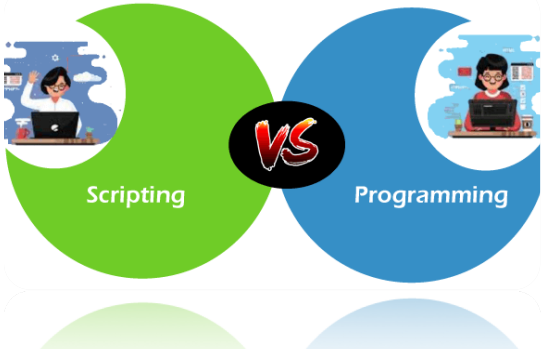
- Interpreted languages: Code in interpreted languages is executed line by line by an interpreter.

  Ex) Python, JavaScript, PHP, and Ruby.

Types of Programming Languages (Cont.)

4) Scripting Languages

- These are often used for automating tasks and quick development.
- These languages are characterized by their ease of use, flexibility, and suitability for quick development and automation
- Python (for scripting and data analysis) and Bash (for shell scripting).

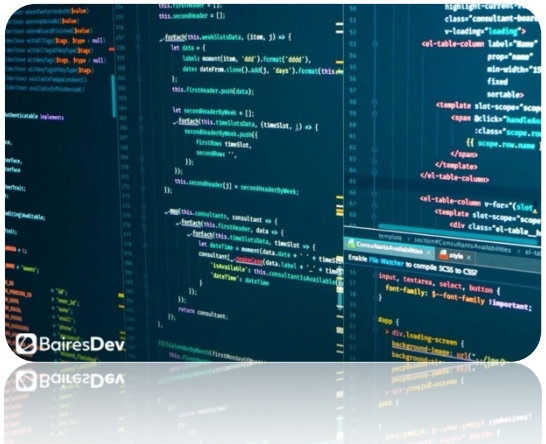


Types of Programming Languages (Cont.)

5) Specialized Languages

Some languages are designed for specific purposes.

- SQL: For managing relational databases.
- R: For statistical analysis.
- MATLAB: For mathematical computing.



Programming Objectives

A program should solve a problem and should be :

- **Correct** : It actually solves the problem
- **Efficient**: Without wasting time or space
- **Readable**: Understandable by another person
- **User-friendly** : In a way that is easy for its user to use

How to write a program?

Steps to writing a program:

Step 1. Think about it

Step 2. Organize your thoughts

Step 3. Write them down in a keywords

Step 4. Translate them into code

How to learn programming?

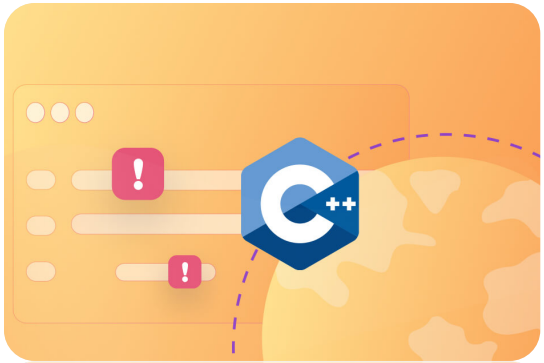
- The best way to learn how to program is to read, then practice, practice and practice

- Working till 3:00 am in the morning on a course work only to find that you typed "==" instead of "=" is a great learning experience

What Language are we going to use?

You are going to learn C++.

- C++ is a powerful general-purpose programming language. It can be used to develop operating systems, browsers, games, and so on.

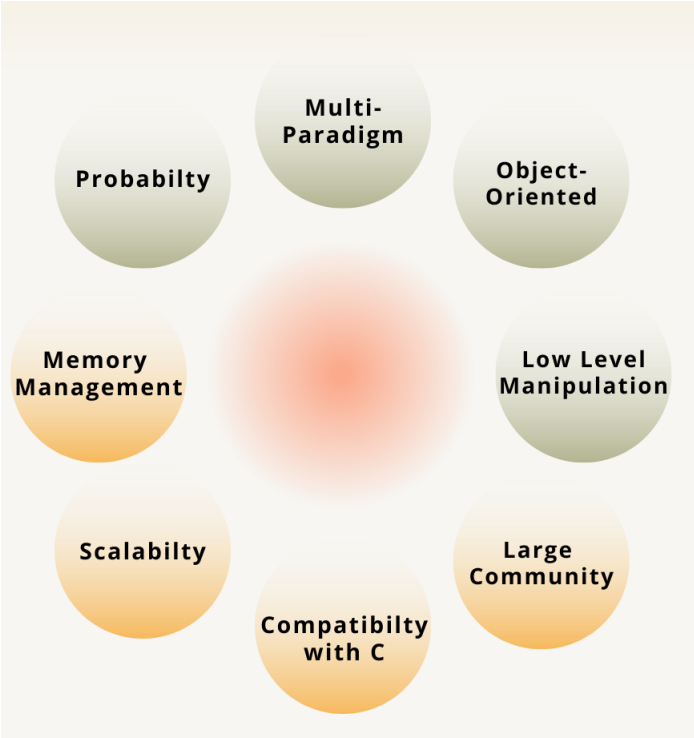


Who Developed C++?

- C++ an extension of C, was developed by Bjarne Stroustrup in the early 1979s at Bell Laboratories.
- The first edition of his book “The C++ Programming Language” was released in 1985.
- C++ is an extension of the C programming language with additional features, including classes, objects, and other features supporting object-oriented programming.

Advantages of learning C++ language

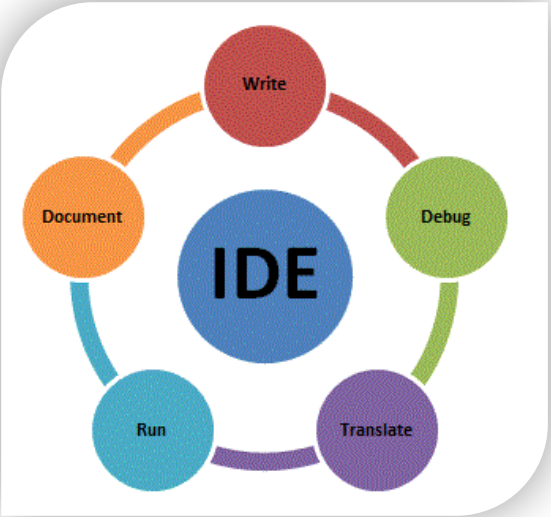
- International standard
- General purpose
- Powerful yet efficient
- Easy to move from C++ to other languages but often not in other direction
- Some other popular languages inspired by C++
- It is FAST



IDE (Integrated Development Environment)

IDE’s provide comprehensive tools for

- Writing and editing codes
- Adding and editing resources
- Building (compiling and linking)
- Debugging codes
- Deploying applications



IDE (Cont.)

IDE's for C++

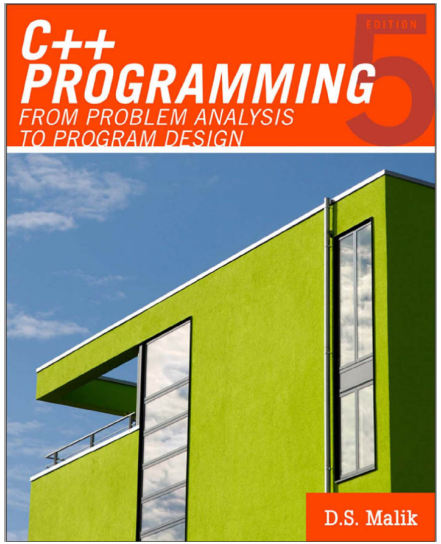
- Microsoft Visual Studio (Community Ed.)
  - Windows OS
- Microsoft Visual Studio Code
  - Mac OS, Linux OS

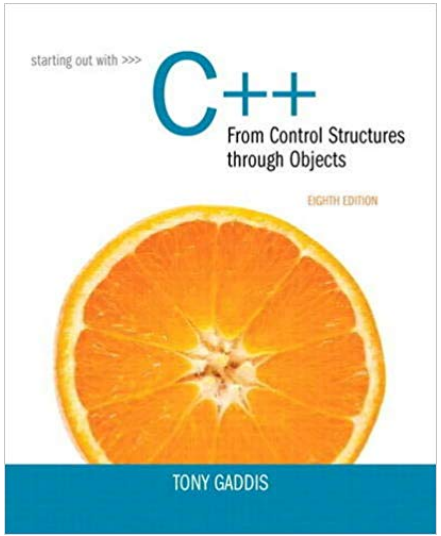
https://visualstudio.microsoft.com

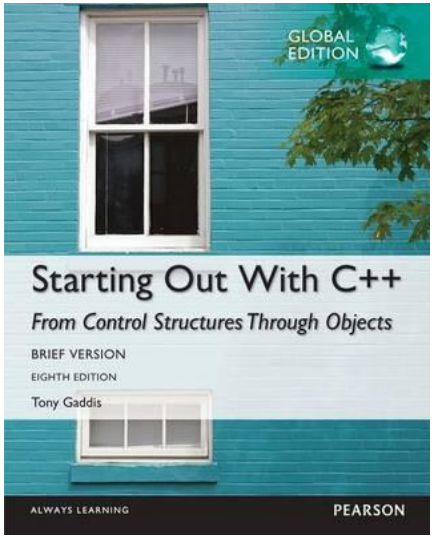
Books

Tony Gaddis, Starting Out with C++: From Control Structures through Objects, 8th Edition

D.S. Malik, C++ PROGRAMMING: From Problem Analysis to Program Design, 5th Edition







References

- **What is Computer Programming**
  https://www.freecodecamp.org/news/what-is-programming/#:~:text=A%20computer%20program%20consists%20of,program%20should%20be%20carried%20out.
- **Advantages of C++**
  https://logicmojo.com/cpp-interview-question
- **Types of Programming Languages**
  https://www.indeed.com/career-advice/career-development/types-of-programming-languages

Part #2

Introduction to C++

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6
7      return 0;
8  }
```

Introduction to C++

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6
7      return 0;
8  }
```

We are putting our code here

Every C++ program has the same essential format.

First Program in C++: Printing a Line of Text

```
1  // The first C++ program
2  #include <iostream>
3  using namespace std;
4  int main() {
5      cout<< "Welcome to TIU";
6      return 0;
7  }
```

Comments

Single-line comments:

```
1  // The first C++ program
```

The // marks the beginning of a *comment*. The compiler ignores everything from the double slash to the end of the line. That means you can type anything you want on that line and the compiler will never complain! Although comments are not required, they are very important to programmers.

Comments

Multi-line comments:

```
/*
    This is a multi-line comment.
    It can put multiple lines.
*/
```

Multi-line comments begin with /* and end with */. Everything between these delimiters is treated as a comment, and it can span multiple lines.

Comments

- Heading
- Author
- Purpose
- Usage
- References
- File Formats
- Restrictions
- Revision History
- Error Handling
- Notes
- Anything else that’ s useful

Comments - Example

```
1  /* ****************************************************************
2   * Program: Assignment.cpp                                        *
3   * Programmer:  Hemin F.                                          *
4   * Purpose: This program for a simple game called seek and find   *
5   * Time and Date: 30.04.2010  02:00 am                            *
6   ****************************************************************/
```

Comments - Example

```
/* ************************************************************
* Program: Assignment.cpp                                     *
* Programmer:  Hemin F.                                       *
* Purpose: This program for a simple game called seek and find *
* Time and Date: 30.04.2010  02:00 am                         *
************************************************************/


# include "stdafx.h"  //include a header file
# include <iostream> //include library
// Get the current calendar time I found it from http://www.cplusplus.com/reference/clibrary/ctime/time
#include <time.h>
using namespace std; //to look in the std library to find the class
int lucky_random(); //prototype of function lucky_random
int UnLucky_random();  //prototype of function Unlucky_random
int conv(int);  //prototype of function conv
char User_Letter; // Global variable

int main()  //main function and mandatory
{
    //generates a new random set each time
    srand(time(0));
    // define a Main Array it has lucky & Unlucky Number
    int Main_Array[5][5];
    //Define this array just to incorrect selection
    int Incorrect_Selection[5][5];
int i,j; //declare i,j as integer using in for loops
int b=65;// you need it to get Alphabet letter
//Start to Create 2-D array and print it
for (i=0;i<5;i++){ //number of rows from 0-4
```

Comments

- You must always comment your programs.
- Comments help you to organize your thoughts.
- Comments help you to remember what you did .
- Comments help the other programmers to understand your program.

First Program in C++: Printing a Line of Text

```
2  #include <iostream>
```

Because this line starts with a #, it is called a *preprocessor directive*. The preprocessor reads your program before it is **compiled**.

The #include directive causes the preprocessor to include the contents of another file in the program. The word inside the brackets, iostream, is the name of the file that is to be included. The iostream file contains code that allows a C++ program to **display output on the screen and read input** from the keyboard.

Declares a set of functions for standard Input/Output

First Program in C++: Printing a Line of Text

```
3  using namespace std;
```

allows you to use elements from the standard C++ library (Standard Template Library - STL) without having to explicitly specify the namespace every time you use a component from it.

The C++ Standard Library provides a collection of classes and functions that are part of the C++ language specification. (Cout, cin)

First Program in C++: Printing a Line of Text

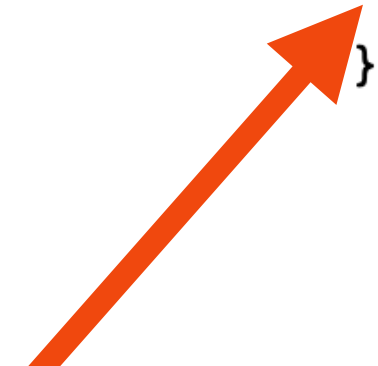
```
3   using namespace std;
```

```
#include <iostream>
using namespace std;

int main() {
   cout << "Hello Tishk!";
   return 0;
}
```

```
#include <iostream>

int main() {
   std::cout << "Hello World!";
   return 0;
}
```



First Program in C++: Printing a Line of Text

```
4  int main() {
5
6
7      return 0;
8  }
```

Main function is a required part of every C++ program. The program execution starts from the main function.

The **int** before main indicates that the function returns an integer. The main function conventionally returns an integer, where a return value of 0 typically indicates successful execution.

First Program in C++: Printing a Line of Text

The Curly braces { and } are used to define blocks of code, which are often referred to as compound statements or blocks.

First Program in C++: Printing a Line of Text

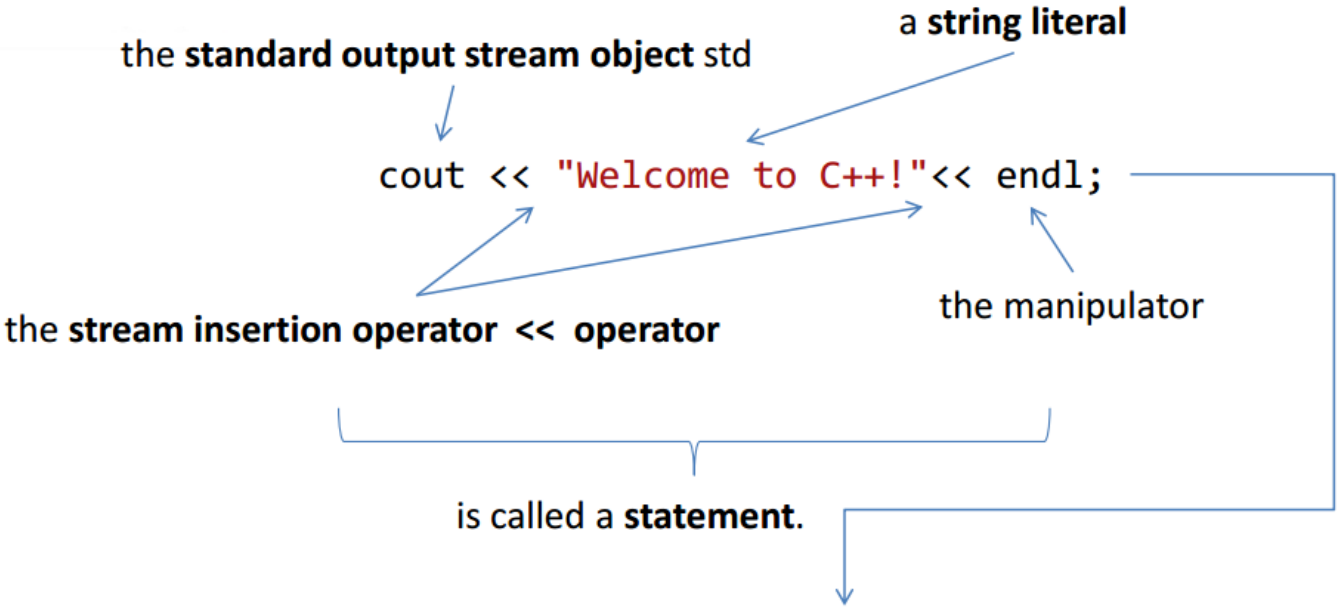
```
5       cout<< "Welcome to TIU";
```

“**cout**”: Stands for "character output" and is an instance of the ostream class, which is part of the C++ Standard Library's input/output stream functionality. It is used to output data to the standard output stream, typically the console.

“**<<**“: Stream Insertion Operator, It is used to insert data into the output stream.

“**;**”: At the end of the line is a semicolon. Just as a period marks the end of a sentence, a semicolon marks the end of a complete statement.

cout in detail



Every C++ statement must end with a **semicolon** (also known as the **statement terminator**). Preprocessor directives (like #include) do not end with a semicolon.

Exactly one function in every C++ program must be named **main**.

In addition, C++ is case sensitive. That means main, Main, MAIN, maiN all are treated the different.

Example

```
1 // A simple C++ program
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7    cout << "Programming is " << "great fun!";
8    return 0;
9 }
```

Program Output

```
Programming is great fun!
```

Example

```
1  // A simple C++ program
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "Programming is ";
8     cout << "great fun!";
9     return 0;
10 }
```

Program Output

```
Programming is great fun!
```

Example

```
1  // An unruly printing program
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "The following items were top sellers";
8     cout << "during the month of June:";
9     cout << "Computer games";
10    cout << "Coffee";
11    cout << "Aspirin";
12    return 0;
13 }
```

Program Output

```
The following items were top sellersduring the month of June:Computer
gamesCoffeeAspirin
```

Example

```
1  // A well-adjusted printing program
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "The following items were top sellers" << endl;
8     cout << "during the month of June:" << endl;
9     cout << "Computer games" << endl;
10    cout << "Coffee" << endl;
11    cout << "Aspirin" << endl;
12    return 0;
13 }
```

Special Characters

Table 2-1 Special Characters

Character	Name	Description
`//`	Double slash	Marks the beginning of a comment.
`#`	Pound sign	Marks the beginning of a preprocessor directive.
`< >`	Opening and closing brackets	Encloses a filename when used with the `#include` directive.
`( )`	Opening and closing parentheses	Used in naming a function, as in `int main()`
`{ }`	Opening and closing braces	Encloses a group of statements, such as the contents of a function.
`" "`	Opening and closing quotation marks	Encloses a string of characters, such as a message that is to be printed on the screen.
`;`	Semicolon	Marks the end of a complete programming statement.

Special Characters (Escape Characters)

There are some certain characters in C++ which can be used with the escape sequence (\) (escape characters or backslash (\) keys).

Control characters:

- \n = Newline
- \b = Backspace
- \t = Horizontal tab
- \r = Return

Punctuation characters:

- \" = Double quote
- \' = Single quote
- \\ = backslash

Escape Characters - Example

```cpp
#include <iostream>
using namespace std;

int main(){
    // Newline
    cout << "Hello" << '\n'
         << "Kurdistan!" << endl;

    // Backspace
    cout << "Hello\bKurdistan!" << endl;

    // Horizontal tab
    cout << "Name\tAge\tCity" << '\n';
    cout << "Alan\t18\tHawler" << '\n';
    cout << "Karzan\t20\tSlemani" << endl;

    // Return
    cout << "12345\rABCD" << endl;

    return 0;
}
```

Output

```
Hello
Kurdistan!
HellKurdistan!
Name    Age     City
Alan    18      Hawler
Karzan  20      Slemani
ABCD5
```

Escape Characters - Example

```cpp
#include <iostream>

using namespace std;

int main(){
    // Double quote
    cout << "This is a double quote: \"Hello\"" << endl;

    // Single quote
    cout
        << "This is a single quote: 'Hello'" << endl;

    // Backslash
    cout << "This is a backslash: \\\\" << endl;

    return 0;
}
```

Output

```
This is a double quote: "Hello"
This is a single quote: 'Hello'
This is a backslash: \\
```

Thank You