C++

1. Introduction

- Pandav Patel

What is C++?

- Object Oriented Programming Language, developed by Bjarne Stroustrup in early 1980s
- Standardized in 1997, ANSI/ISO standard committee
- Most C programs are valid C++ programs
- Addition of OOP is the major change from C language

Applications of C++

- Versatile language
- System programming – Parts of Linux and Windows
- Application programming – Paint, Notepad etc.

- Now a days, very popular for competitive coding due to its good computing performance compared to java and python and avaibility of very good library compared to C language.

Sample C++ program

```
#include<iostream>
using namespace std;
/* Comments same as C Language */
int main() {
     // Like C language, execution starts from main
    cout << "Hello World!\n";
    return 0;
}
```

Explanation of sample program

- **iostream file inclusion**
  - No .h extension – old style don’t use it
  - For C library files prepend library name with ‘c’ and drop .h extension
  - e.g. **#include<cstdio>** instead of **#include<stdio.h>**
- **using and namespace keywords**
  - Will be covered in later lectures in more details
- **cout**
  - cout is output stream object declared in iostream file
  - It represents standard output
  - << is called **insertion** or **put to** operator declared in iostream
    - Writes contents of variable/constant to output

Another C++ program

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

int main() {
    int num1, num2;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

    int sum = num1 + num2; // Variable can be declared anywhere

    cout << "Sum is: " << sum << endl; // No need of format specifier

    return 0;
}
```

Explanation of the program

- **cin**
  - input stream object declared in iostream file
  - It represents standard input
  - >> is **Extraction** or **Get From** operator declared in iostream
  - Reads data from input in to variables
- **Operator overloading** – bitwise shift Vs insertion/Extraction
  - Will be more clear later when we overload operators ourselves
- Cascading of I/O operators
- **endl** is manipulator defined in iostream
  - Inserts new line to the output and flushes output buffer

Compiling and Execution of c++ program

- **Compilation**
  - $**g++ prog.cpp**
- **Execution**
  - $**./a.out**
- IDE like Code::Blocks, VS Code, Visual Studio, Eclipse can be used to make compiling and debugging easier.

C Vs C++ Sample Program

```
#include<stdio.h>

int main()
{
    int num1, num2, result;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    result = num1 + num2;

    printf("Result is %d\n", result);

    return 0;
}
```

```
#include<cstdio>

int main()
{
    int num1, num2, result;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    result = num1 + num2;

    printf("Result is %d\n", result);

    return 0;
}
```

C Vs C++ Sample Program

```
#include<cstdio>

int main()
{
    int num1, num2, result;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    result = num1 + num2;

    printf("Result is %d\n", result);

    return 0;
}
```

```
#include<cstdio>
#include<iostream>

using namespace std;

int main()
{
    int num1, num2;

    cout << "Enter two numbers: ";
    cin >> num1;
    scanf("%d", &num2);

    int result = num1 + num2;

    printf("Result is %d\n", result);

    return 0;
}
```

C Vs C++ Sample Program

```
#include<cstdio>
#include<iostream>

using namespace std;

int main()
{
	int num1, num2;

	cout << "Enter two numbers: ";
	cin >> num1;
	scanf("%d", &num2);

	int result = num1 + num2;

	printf("Result is %d\n", result);

	return 0;
}
```

```
#include<iostream>

using namespace std;

int main()
{
	int num1, num2;

	cout << "Enter two numbers: ";
	cin >> num1 >> num2;

	int result = num1 + num2;

	cout << "Result is " << result << "\n";

	return 0;
}
```

Interesting reads...

- How to improve I/O speed?
    - ios_base::sync_with_stdio(false);
        - disables the synchronization between the C and C++ standard streams
        - Makes I/O faster
    - cin.tie(NULL);
        - unties cin from cout, makes I/O faster
        - If cin and cout are tied, output is flushed before reading input each time
- "\n" Vs endl
    - "\n" does not flush the output stream but endl does flush output stream