

SESSION 02: BASIC COMPUTER PROGRAMMING IN C++ (1)

Introduction to Programming in C++

C++ is a highly efficient and flexible programming language that forms the basis for many modern software technologies. This session aims to introduce you to the fundamentals of programming in C++.

Objectives of This Session

- Understand the basic structure of a C++ program.
- Learn about variables, data types, and basic input/output.
- Write simple C++ programs to reinforce understanding.

1. Overview of C++

- **History and Significance:** Developed by Bjarne Stroustrup in 1985, C++ is known for its performance and is widely used in system/software development, game programming, and real-time simulation.
- **Characteristics of C++:** It is a statically typed, free-form, multi-paradigm, compiled language that supports procedural, object-oriented, and generic programming.

2. Setting Up the Development Environment

- IDEs and Compilers:

- GCC (GNU Compiler Collection)
- Visual Studio Code with C++ extension
- Dev-C++

- Installation Guides:

- [Dev C++ Setup]

(www.geeksforgeeks.org/how-to-download-and-install-dev-c-on-windows/)

- [Visual Studio Code Setup] (<https://code.visualstudio.com/docs/languages/cpp>)
- [Code With Harry for learning C++] (<https://bit.ly/4b3XLUy>)

3. Structure of a C++ Program

- Basic Structure:

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

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

- Explanation:

- `#include<iostream>`: Includes the Standard Input Output library.
- `using namespace std;`: Standard namespace.
- `main()`: The main function where execution begins.
- `cout`: Standard character output stream.
- `return 0;`: Ends the program.

4. Basic Concepts in C++

- Variables and Data Types:

- Integers, characters, floating-point numbers, and booleans.
- Variable declaration and initialization: `int age = 30;`

- Input/Output Operations:

- Input using `cin`: `cin >> age;`
- Output using `cout`: `cout << "Age: " << age;`

5. Writing Your First Program

- Program to Add Two Numbers:

```
#include<iostream>  
using namespace std;  
  
int main() {  
    int num1, num2, sum;  
    cout << "Enter two numbers: ";  
    cin >> num1 >> num2;  
    sum = num1 + num2;  
    cout << "Sum = " << sum;  
    return 0;  
}
```

6. Practical Tips

- **Debugging Tips:** Use `cout` to print variable values to debug.
- **Practice Problems:**

- Write a program to calculate the area of a rectangle.
- Create a simple calculator that performs basic arithmetic operations.

7. Getting Started with Learning

- Resources:

- [Code With Harry for learning C++] (<https://bit.ly/4b3XLUy>)
- [W3 Schools for learning C++] (<https://www.w3schools.com/cpp/>)

Conclusion

This session provides an introduction to the fundamental concepts of programming in C++. Understanding these basics is crucial for your journey in learning more complex aspects of the language.

OFF THE SCHOOL