CSE 232

Programming with C++

Lecture 1

Course Overview

Prepared by

Prof. Dr. Md. Mijanur Rahman
Dept. of Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University, Bangladesh
Email: mijan@jkkniu.edu.bd
Web: mijanrahman.com

Contents

Course Overview

This Course

- "Programming with C++" is a foundational course designed to teach the fundamentals of the C++ programming language.
- This course is often part of computer science or engineering curricula and is suitable for beginners who are new to programming as well as those who have some prior experience with other languages.

- Key Topics Covered: Introduction to C++, Data Types and Variables, Control Structures, Functions, Arrays and Strings, Object-Oriented Programming (OOP) Concepts, Operator Overloading, File Handling.

Course Outcomes

- By the end of the course, students should be able to:
  - Write, compile, and debug C++ programs.
  - Understand and apply fundamental programming concepts.
  - Utilize object-oriented programming principles to design and implement software.
  - Work with file I/O for basic data storage and retrieval.

Class Duration

- The duration of each semester will be 19 weeks whose breakdown is as follows:

Class	14 weeks
Recess before Semester Final Examination	2 weeks
Semester Final Examination	3 weeks
Total	19 weeks

- For theoretical course (Credit 3.0), three hours class will be conducted in a week. For lab course (Credit 1.5), three hours practical class will be conducted in a week

Course Evaluation

- **Evaluation System for Theoretical Course -** The marking and student evaluation system will be as follows:

1. Continuous Assessments : 40%
   a. Class attendance : 10%
   b. Midterm Exam-1 : 10%
   c. Midterm Exam-2 : 10%
   d. Midterm Exam-3 : 10%
2. Semester Final Exam : 60%
3. **Total** **: 100%**

Course Evaluation

- **Evaluation System for Lab Course -** The marking and student evaluation system will be as follows:

1. Continuous Assessments : 40%
   a. Class attendance : 10%
   b. Lab Report : 10%
   c. Continuous Evaluation : 20%
2. Lab Final Exam : 60%
   a. Viva Voce : 20%
   b. Lab Test : 30%
   c. Ans. Script : 10%
3. **Total : 100%**

Class Attendance

- The distribution of marks for class attendance (theoretical and practical) will be as follows:

Attendance	Marks
90% and above	10
85% to 89%	09
80% to 84 %	08
75% to 79 %	07
70% to 74%	06
65% to 69%	05
60 % to 64%	04
55% to 59 %	03
50% to 54%	02
Less than 50%	00

- A student shall have to attend at least 75% of theoretical and practical classes held in a course.
- In case of shortage of attendance (not bellow 60%), student will be allowed to sit for examination after paying of taka 500/- as irregular fee for each course in university account.
- Below 60% will NOT be allowed to sit for examination.

The Grading System

- Letter grades and corresponding grade points will be awarded in accordance with the provisions shown below:

Numerical Grade	Letter grade	Grade Point	Interpretation
80% and above	A+	4.00	**Outstanding**
75% to less than 80%	A	3.75	**Excellent**
70% to less than 75%	A-	3.50	**Very Good**
65% to less than 70%	B+	3.25	**Good**
60% to less than 65%	B	3.00	**Satisfactory**
55% to less than 60	B-	2.75	**Nearly Satisfactory**
50% to less than 55%	C+	2.50	**Average**
45% to less than 50%	C	2.25	**Nearly Average**
40% to less than 45%	D	2.00	**Poor**
Less than 40%	**F**	**0**	**Fail**

Course Contents

- **CSE 232: Programming with C++**

Credits: 1.5; Full Marks: 100; Final Exam Time: 6.0 Hours

- **Course Syllabus:**
  - Introduction to C++ programming language: history, and features, basic syntax, data types, variables, input/output operations.
  - Control structures: if-else, switch-case, loops (while, do-while, for), logical and relational operators.
  - Introduction to functions: function prototypes, defining and calling functions, pass-by-value and pass-by-reference, recursion.
  - Arrays in C++: declaration, initialization, accessing elements, multi-dimensional arrays.
  - Pointers: basics, pointer arithmetic, dynamic memory allocation, arrays vs. pointers, pointers and arrays.
  - Introduction to object-oriented programming: classes and objects, encapsulation, constructors and destructors, Inheritance, base and derived classes, access specifiers, polymorphism, function overriding, virtual functions, operator overloading, friend functions, static members, namespaces, exception handling.
  - File handling in C++: reading from and writing to files, file streams, error handling.
  - Advanced C++ Topics and Project Work: Templates, template classes and functions, project work and presentations.

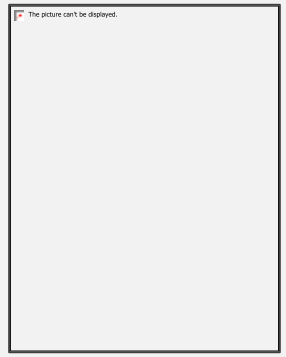
Course Objectives

- The course objectives of "Programming with C++" are designed to ensure that students gain a solid understanding of both the theoretical and practical aspects of C++ programming. The key objectives are:
  - Introduce students to fundamental programming concepts such as variables, data types, control structures (loops, conditionals), functions, and basic input/output operations in C++.
  - Teach the principles of object-oriented programming using C++. This includes classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
  - Teach file input/output operations in C++ including reading from and writing to files, handling file streams, and error handling.
  - Develop students' problem-solving abilities through programming assignments, projects, and exercises that require applying C++ concepts to real-world problems.
  - Provide opportunities for students to work on larger programming projects individually or in teams, allowing them to apply their knowledge of C++ to develop complete software applications.

Text Books and References

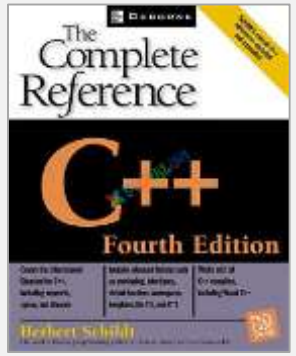
- **Online Course Materials:**
  - https://www.javatpoint.com/cpp-tutorial
  - https://www.geeksforgeeks.org/c-plus-plus/
- **Online C Compiler:**
  - https://www.programiz.com/cpp-programming/online-compiler/
- **Software (Code:: Blocks):**
  - https://www.codeblocks.org/downloads/

Text Books:

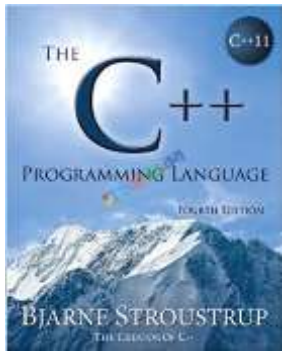


Programming & Software Development
With AI and Machine Learning Concepts
By M. M. Rahman

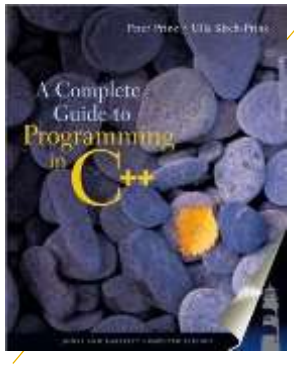
Object Oriented programming with C++
By E. Balagurusamy



The Complete Reference C++
By Herbert Schildt



The C++ Programming Language
By Bjarne Stroustrup



A Complete Guide to Programming in C++
By Ulla Kirch-Prinz

Course Prerequisites and Dependencies

- Basic understanding of computers, mathematics, and basic computer programming.
- Some prior programming experience (in languages like C) can be beneficial but is not always required.
- The course teaching language is **English**, so students have to have communication, reading and apprehension skills of English

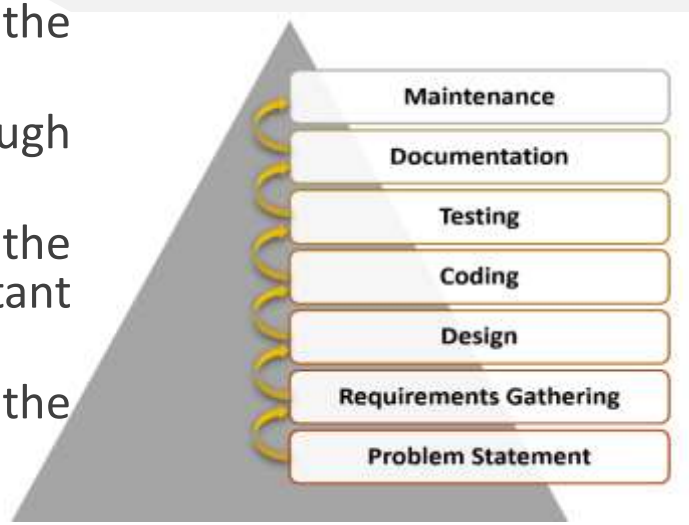
- This course serves as a gateway to more advanced topics in software development and other programming languages.

Computer Programming

- Computer programming is the process of designing and building software applications, tools, and systems.
- It involves writing code in a programming language, testing and debugging it, and then integrating it into a larger software system.
- Programming languages are used to write computer programs.
- The most popular programming languages used today:
  - Python, Java, JavaScript, C/C++, C#, PHP, HTML, SQL, Prolog, Swit, Ruby, Go, etc.

Software Development Stages

- The major stages are summarized below:
  1. **Problem statement:** Identifying and analyzing the problem that the software is intended to solve.
  2. **Requirements gathering:** Understanding the requirements of the software system and what it needs to accomplish.
  3. **Design:** Creating a high-level design of the software system, including the algorithms, data structures, and overall architecture.
  4. **Coding:** Writing the actual code in a programming language, following the design.
  5. **Testing:** Debugging the code and ensuring that it works correctly through the use of test cases and other techniques.
  6. **Documentation:** Creating and maintaining documents that describe the software system, its architecture, design, functionality, and other important aspects.
  7. **Maintenance:** Updating and fixing the code as needed over time to keep the software system running smoothly.



Programming Languages…

- Programming languages can be divided into several categories, including:
  - **Procedural programming languages:** These languages use a procedural approach to programming, where a series of steps or procedures are defined to solve a problem. Examples of procedural programming languages include C, Pascal, and Fortran.
  - **Object-oriented programming languages:** These languages use an object-oriented approach to programming, where data is organized into objects, and the behavior of these objects is defined by the methods that they contain. Examples of object-oriented programming languages include Java, Python, and Ruby.
  - **Functional programming languages:** These languages use a functional approach to programming, where functions are first-class citizens and program execution is based on evaluating mathematical functions. Examples of functional programming languages include Haskell, Lisp, and Scheme.

Programming Languages

- **Scripting languages:** These languages are typically interpreted rather than compiled and are used for scripting and automating tasks. Examples of scripting languages include Perl, JavaScript, and Ruby.
- **Logic Programming languages:** Logic programming is a type of programming paradigm based on formal logic. It is used to specify relationships between objects and to define rules that can be used to deduce new information from existing information. The most well-known logic programming language is Prolog (Programming in Logic).
- **Low-level programming languages:** These languages are closer to machine language and provide direct control over computer hardware. Examples of low-level programming languages include Assembly and C.
- **High-level programming languages:** These languages provide a high-level abstraction from computer hardware and are designed to be easier for humans to read and write. Examples of high-level programming languages include Python, Java, and Ruby.

Object-Oriented Programming (OOP)…

- Object-oriented programming (OOP) is a programming paradigm that is based on the concept of objects, which are instances of classes that represent real-world entities.
- In an object-oriented programming language, the program is organized around objects, and objects communicate with each other by sending messages.

Object-Oriented Programming (OOP)

- **The key features of object-oriented programming languages include:**
- **Encapsulation:** Encapsulation is the mechanism that binds together the data and functions that operate on that data within an object.
- **Inheritance:** Inheritance is a mechanism that allows objects to inherit characteristics and behaviors from parent classes. This allows for the creation of new classes based on existing classes, making it easier to reuse code and reduce duplicated efforts.
- **Polymorphism:** Polymorphism is the ability of an object to take on many forms. This can be achieved through method overriding, where a subclass provides a new implementation of a method defined in its parent class, or through method overloading, where a single method can have multiple implementations based on the number or type of its arguments.
- **Dynamic Dispatch:** Dynamic dispatch is the mechanism by which a method is chosen for execution at runtime based on the type of object that it is called on.
- **Class-based:** Object-oriented programming is class-based, which means that objects are instances of classes, and classes define the data and behaviors that objects have.

Lecture 1

Course Overview

THE END