

| PART A: Introduction | | | |
|--|--|--|------------------------|
| Program: Certificate | | Class: UG | |
| Year: 1 Year | | Session: 2025-26 | |
| Subject: Computer Application | | | |
| 1 | Course Code | C3 | |
| 2 | Course Title | Object Oriented Programming using C++ | |
| 3 | Course Type (Core Course/Elective/Generic Elective/ Vocational | Core Course | |
| 4 | Pre-Requisite (if any) | To study this course, a student must have had the class. | |
| 5 | Course Learning Outcomes(CLO) | On completion of this course, learners will be able to: • Learn the principles of object-oriented programming and apply them in C++. • Develop structured, reusable, and modular applications. • Implement file handling, exception handling, and STL. • Design optimized and scalable applications using C++. • Integrate structured problem-solving techniques with traditional Indian logic. | |
| 6 | Credit Value | Theory-4 Credits | |
| 7 | Total Marks | Max. Marks: 30+70 | Min. Passing Marks: 35 |
| PART B: Content of the Course | | | |
| No. of Lectures (in hours per week): 2 Hrs. per week | | | |
| Total No. of Lectures: 60 Hrs. | | | |
| Module | Topics | | No. of Lectures |
| I | Indian Knowledge System (IKS) and Programming Concepts Ancient Indian contributions to mathematics and logic (Aryabhata, Brahmagupta) and their relevance to computational models. Concepts of structured problem-solving in Indian logic (Nyaya, Mimansa) and their influence on algorithmic thinking. Traditional knowledge classification systems and their parallels in modern databases and file structures. Application of IKS in Object-Oriented Programming: • Data Structures & Algorithms: Influence of Indian mathematics on recursion and sorting techniques. • Logical Reasoning in Programming: Vedic methods for structured reasoning applied in decision-making constructs. • Code Efficiency & Optimization: Ancient computational efficiency principles and their application in modern programming practices. Keywords: Indian Knowledge System, Vedic Mathematics, Ancient Algorithms, Computational | | 10 |

| | | |
|-----|--|----|
| | <p>Thinking, Logic Systems.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Research ancient Indian algorithmic techniques and implement a sorting algorithm using them. • Create programs simulating Vedic computational methods. • Develop a project showcasing logic systems inspired by Nyaya or Mimamsa. • Apply traditional data classification concepts to a small database project. | |
| II | <p>Introduction to Object-Oriented Programming & C++ Basics</p> <p>Introduction to programming paradigms: Procedural vs. Object-Oriented Programming; Features and benefits of Object-Oriented Programming (OOP).</p> <p>Basics of C++ programming: Syntax, data types, variables, constants, operators, expressions, input/output handling using cin and cout.</p> <p>Control structures: Decision-making statements (if, if-else, switch), looping constructs (for, while, do-while).</p> <p>Functions in C++: Function declaration, definition, function overloading, inline functions, recursion.</p> <p>Keywords: OOP, Procedural vs. Object-Oriented, C++ Syntax, Variables, Operators, Loops, Functions.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Write a C++ program using basic syntax, cin, and cout. • Demonstrate looping constructs with nested loops. • Create a recursive function to calculate factorial or Fibonacci series. • Implement a simple C++ program with function overloading. | 10 |
| III | <p>Classes, Objects, and Memory Management Concept of a class and object: Defining a class, creating objects, accessing class members.</p> <p>Constructors and destructors: Types and uses, this pointer. Static data members and static member functions: Friend functions and friend classes.</p> <p>Dynamic memory allocation: Using new and delete.</p> <p>Other concepts: Array of objects, object as function arguments, returning objects from functions.</p> <p>Keywords: Classes, Objects, Constructors, Destructors, Static</p> | 10 |
| | <p>Members, Friend Functions, Dynamic Memory.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Create a C++ program with multiple classes and objects. • Implement constructor overloading in a C++ project. • Use dynamic memory allocation to create and manage an array | |

| | | |
|----|--|----|
| | <p>of objects.</p> <ul style="list-style-type: none"> • Write a program demonstrating friend functions and static members. | |
| IV | <p>Inheritance, Polymorphism, and Operator Overloading Concept of inheritance: Types of inheritance: single, multiple, multilevel, hierarchical, hybrid, visibility modes in inheritance, function overriding, virtual base class.</p> <p>Polymorphism:</p> <ul style="list-style-type: none"> • Compile-time polymorphism: Function overloading, operator overloading. • Runtime polymorphism: Virtual functions, pure virtual functions, abstract classes. <p>Operator overloading: Rules for operator overloading, overloading unary and binary operators, type conversion.</p> <p>Keywords: Inheritance, Polymorphism, Virtual Functions, Function Overloading, Operator Overloading.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Implement single and multiple inheritance in a C++ program. • Create an application demonstrating function overloading and overriding. • Use operator overloading for arithmetic operations on custom objects. • Develop a polymorphic base class with derived classes implementing different functionalities. | 10 |
| V | <p>File Handling and Advanced C++ Features File stream classes in C++: Reading/writing files, file modes, file pointers, error handling in file operations.</p> <p>Templates:</p> <ul style="list-style-type: none"> • Function templates and class templates • Advantages of generic programming <p>Exception handling: Using try, catch, throw, and handling multiple exceptions.</p> <p>Standard Template Library (STL):</p> <ul style="list-style-type: none"> • Containers: vector, list, map, set • Iterators • Algorithms <p>Keywords: File Handling, Templates, Exception Handling, STL, Containers, Iterators.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Write a program to read and write data to a file. • Create a template-based sorting function. • Demonstrate exception handling with multiple catch blocks. | 10 |
| | <ul style="list-style-type: none"> • Use STL containers and iterators to build a small data structure. | |

| | | |
|---|---|----|
| VI | <p>Applications of C++ and Industry Practices Real-world applications of C++: System programming, game development, financial applications, embedded systems.</p> <p>Introduction to GUI programming with C++: Basics of Qt.</p> <p>Debugging techniques: Profiling tools and code optimization methods.</p> <p>Capstone project: Building an application using OOP concepts.</p> <p>Keywords: C++ Applications, GUI, Debugging, Optimization, Capstone Project.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Create a mini C++ project with GUI using Qt framework. • Demonstrate code optimization techniques in a C++ program. • Use debugging tools to identify and fix errors in a C++ project. • Design a financial calculator or inventory management system. | 10 |
| PART C: Learning Resources | | |
| Textbooks, Reference Books, Other Resources | | |
| <p>Suggested Readings:</p> <ol style="list-style-type: none"> 1. The C++ Programming Language – Bjarne Stroustrup (Addison-Wesley). 2. C++ Primer – Stanley B. Lippman, Josée Lajoie, Barbara E. Moo (Pearson). 3. Effective C++ – Scott Meyers (O'Reilly). 4. Accelerated C++ – Andrew Koenig, Barbara E. Moo (Addison-Wesley). 5. Programming: Principles and Practice Using C++ – Bjarne Stroustrup. 6. Object-Oriented Programming in C++ – Robert Lafore. <p>Suggestive digital platform web links:</p> <ol style="list-style-type: none"> 1. https://cplusplus.com/doc/tutorial/ – C++ Documentation and Tutorials. 2. https://learn-cpp.com/ – Learn C++ with hands-on examples. 3. https://cplusplus.com/ – C++ Programming from basics to advanced concepts. 4. https://cplusplus.com/ – C++ problem-solving and implementation. 5. https://nptel.ac.in/courses/106/105/106105151 – NPTEL Online Course on Programming in C++. | | |
| Suggested equivalent online courses | | |
| https://nptel.ac.in/courses/106/105/106105163/ | | |
| PART D: Assessment and Evaluation | | |
| <p>Maximum Marks :100</p> <p>Continued Comprehensive Evaluation (CCE):30</p> <p>University Exam(UE):70</p> <p>Time:03:00 Hours</p> | | |

| | | |
|---|--|---|
| Internal Assessment: Continued Comprehensive Evaluation (CCE): | Class Tests/ Presentation / Assignment | 30 Marks |
| External Assessment: University Exam | Section(A):Very Short Questions Section (B) : Short Questions Section (C) : Long Questions | 70 Marks |
| Any remarks/suggestions: Learning's in the course should be emphasized more on practical aspects and real world problems and their solutions. | | |
| PART A: Introduction | | |
| Program: Certificate | Class: UG | Year: 1 Year |
| Session: 2025-26 | | |
| Subject: Computer Application | | |
| 1 | Course Code | C3 |
| 2 | Course Title | Object Oriented Programming using C++ |
| 3 | Course Type (Core Course/Elective/Generic Elective/ Vocational | Core-Course |
| 4 | Pre-Requisite (if any) | To study this course, a student must have had the class. |
| 5 | Course Learning Outcomes(CLO) | On completion of this course, learners will be able to: <ul style="list-style-type: none"> • Learn the principles of object-oriented programming and apply them in C++. • Develop structured, reusable, and modular applications. |
| | | <ul style="list-style-type: none"> • Implement file handling, exception handling, and STL. • Design optimized and scalable applications using C++. |
| 6 | Credit Value | Practical-2 Credits |
| 7 | Total Marks | Max. Marks: 30+70 Min. Passing Marks: 35 |
| PART B: Content of the Course | | |
| No. of Lab. Practical's (in hours per week): 2 Hrs. per week | | |
| Total No. of Labs:30 | | |
| | Suggestive list of Practical's | No. of Labs. |

| | | |
|--|---|-----------|
| | <p>C++ Practical Tasks</p> <ol style="list-style-type: none"> 1. Write a C++ program to demonstrate basic input and output operations using cin and cout. 2. Implement a C++ program to demonstrate data types, operators, and type conversion. 3. Write a program to implement control structures (if-else, switch-case, loops). 4. Implement functions in C++, including function overloading and recursion. 5. Write a C++ program to demonstrate call by value and call by reference. 6. Implement a C++ program to define a class and create objects. 7. Write a program to use constructors and destructors in a class. 8. Implement a program to demonstrate static data members and static functions. 9. Write a program to implement friend functions and friend classes. 10. Implement a C++ program to demonstrate this pointer and dynamic memory allocation. 11. Write a program to implement single inheritance with base and derived classes. 12. Implement a program to demonstrate multiple inheritance. 13. Write a program to implement multilevel inheritance in C++. 14. Implement a C++ program to demonstrate hierarchical inheritance. 15. Write a program to implement hybrid inheritance using virtual base classes. 16. Implement polymorphism by demonstrating function overloading and function overriding. 17. Write a program to implement operator overloading (unary and binary operators). 18. Implement a C++ program to demonstrate virtual functions and pure virtual functions. 19. Write a program to create a template function and a template class. 20. Implement a program to demonstrate exception handling using try, catch, and throw. | <p>70</p> |
| <p>PART C: Learning Resources</p> | | |
| <p>Textbooks, Reference Books, Other Resources</p> | | |

Suggested Readings:

1. The C++ Programming Language – Bjarne Stroustrup (Addison-Wesley).
2. C++ Primer – Stanley B. Lippman, Josée Lajoie, Barbara E. Moo (Pearson).
3. Effective C++ – Scott Meyers (O'Reilly).
4. Accelerated C++ – Andrew Koenig, Barbara E. Moo (Addison-Wesley).
5. Programming: Principles and Practice Using C++ – Bjarne Stroustrup.
6. Object-Oriented Programming in C++ – Robert Lafore.

Suggestive Digital Platform Web Links:

- <https://cplusplus.com/doc/tutorial/> – C++ Documentation and Tutorials.
- <https://learn-cpp.com/> – Learn C++ with hands-on examples.
- <https://www.cplusplus.com/doc/tutorial/> – C++ Programming from basics to advanced concepts.
- <https://www.cplusplus.com/doc/tutorial/> – C++ problem-solving and implementation.
- <https://nptel.ac.in/courses/106/105/106105151> – NPTEL Online Course on Programming in C++.

Suggested equivalent online courses

<https://nptel.ac.in/courses/106/105/106105163/>

PART D: Assessment and Evaluation

Maximum Marks :100

Continued Comprehensive Evaluation (CCE):30

University Exam(UE):70

Time:03:00 Hours

| | | |
|--|--|----------|
| Internal Assessment: Continued Comprehensive Evaluation (CCE): | Class Tests/ Presentation / Assignment | 30 Marks |
| External Assessment: University Exam | Section(A):Very Short Questions Section (B) : Short Questions Section (C) : Long Questions | 70 Marks |

Any remarks/suggestions: Learnings in the course should be emphasized more on real world problems and their solutions.