**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

**LESSION PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| Name of the Course Teacher | J. SEETHARAMAN | Designation, Department, Staff Code | Asst. Prof./ CSA |
| Degree & Branch | I BSc(Cyber Security) | Year / Semester | I / I |
| Course | Object Oriented Programming with C++ | Credits | 3 |

**COURSE OBJECTIVES**

To introduce the basic concepts of Object Oriented Programming.

To acquire knowledge on C++ functions.

To become aware of significant classes and Objects.

To impart knowledge on inheritance and polymorphism concepts in C++.

To explore various file I/O Operations.

**COURSE OUTCOMES**

Understand the basic concepts and need of Object oriented programming.

Explain the various keywords and Tokens in C++.

Describe the various class concepts in C++.

Explain the basic principles of inheritance and polymorphism

Examine the various file I/O operations.

**TEXT BOOKS**

1. Herbert Schildt(2017), "C++ Complete Reference", Fourth edition, TMH,

2. Bjarne Stroustrup, (2013 )“The C++ programming language”, Addison Wesley,

3. Balaguruswamy, “Programming in C++”, 5th Edition, Tata McGraw Hill Education

Private Limited, 2011.

**REFERENCE BOOKS:**

1. Paul J.Deitel , Harvey M.Deitel, “C++: How To Program”, Prentice Hall, 2010

2. Robert Lafore, “Object Oriented programming Using C++”. Waite’s Group, 1999.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit I : OOP Basics (Week 1–3) 12 Hours**  **Learning Objectives:**   * Understand OOP basics. * Learn data types, input/output, and variables/operators in C++.   Introduction to C++ - Principles Of Object Oriented Programming (OOP) – Basic Concepts  of OOP - Benefits of OOP – Applications of OOP – Tokens – Keywords – Identifiers –  Variables – Operators – Manipulators - Expressions. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 1 | Problems and solutions – OOP Approach | Lecture & Discussion | Simulating real time problem’s solutions oop |
| 1 | Computing Real time problems with OOP Principles | Lecture & Discussion | Simulating real time problem’s solutions oop |
| 2 | Introduction to C++ as OOP language | Examples & Practice | Worksheet on C++ OOP blocks |
| 2 | Input and Output Statements | Examples & Practice | Interactive practice |
| 3 | Functional vs oop Programming | Examples & Practice | Mini assignments/study on comparison |
| 3 | Data types, variables, constants, operators | Hands-on & Discussion | Loop-based coding tasks |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit II: C++ Functions and Coding blocks (Week 4–6) – 12 Hours**  **Learning Objectives:**   * Define and use functions. * Understand and apply control statements   Decision Making Statements – Looping Statements - Functions - Main Function - Function  Prototyping – Passing Parameters to Functions - Values Return by Functions – Inline  Functions - Friend Functions. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 4 | Functions: Definition, calling, types, arguments | Lecture + Lab | Write reusable functions |
| 4 | Arrays and String operations | Lecture & Discussion | String manipulation problems |
| 5 | Control statements | Lecture & Discussion | Problem solving with control statements |
| 5 | Control statements | Lecture & Discussion | Problem solving with control statements |
| 6 | Looping statements | Lecture & Discussion | Problem solving with control and looping statements |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit III: OOP blocks (Week 7–9) – 12 Hours**  **Learning Objectives:**   * Use object-oriented programming. * Understand and build solutions with OOP fundamentals.   Classes and Objects - Constructors and Destructors - Types of Constructor - Inheritance –  Types of Inheritance – Function Overloading - Operator Overloading. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 7 | OOP Concepts: Class, Object, Attributes, Inheritance | Theory + Practice | Build class-based programs |
| 7 | Polymorphism, Overloading | Theory + Practice | Code examples |
| 8 | Constructors | Theory + Practice | Code examples |
| 8 | Inheritance | Theory + Practice | Code examples |
| 9 | Operator overloading | Theory + Practice | Code examples |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit IV: Web and GUI Programming (Week 10–12) – 12 Hours**  **Learning Objectives:**   * Pointers and I/O operations. * Understand templates   Pointers - Virtual Functions and Polymorphism - Managing Console I/O operations -  Templates Introduction – Function templates | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 10 | Pointers | Theory + Practice | Code examples |
| 11 | Virtual functions, I/O operations | Theory + Practice | Code examples |
| 12 | Templates | Theory + Practice | Code examples |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit V: Working with Files (Week 13–15) – 12 Hours**  **Learning Objectives:**   * Understand streams and files.   Working with Files – Classes for File Stream Operations – Opening and Closing of a File –  Updating a File - End of File Deduction. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 13 | Streams and file operations | Theory + Practice | Code examples |
| 14 | Reading files | Theory + Practice | Code examples |
| 15 | Writing files | Theory + Practice | Code examples |

**Assessment Strategy:**

* **Internal Assessment**: Periodic class tests, coding assignments, quizzes
* **Practical Exam**: Midterm and Final lab exams
* **Project**: Mini project integrating OOP, file handling, and data analysis

-------------------------------------------------------------------------------------------------------------------------------

**OBJECT ORIENTED PROGRAMMING with C++ LAB.**

**Weekly OOP with C++ Practical Lesson Plan (4 Hours/Week)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Experiment Topic** | **Learning Objectives** | **Lab Activities** | **Assessment** |
| **1,2** | Github Account creation and usage | - Introduction to version control system, github account creation with sample repo and file | -creating github repo with sample repo, reviewers and files | Verification of github repo with sample repo, reviewers and files |
| **3,4** | Classes & Objects | - Understand basic oop building blocks | Program to display Employee details Using Classes and Object. | Verification with object, class instantiation with params |
| **5,6** | Friend function | Friend function | Program to find the Mean Value Using Friend Function. | Program to find the Mean Value Using Friend Function. |
| **7,8** | inline function | inline function | inline function sample program | inline function sample program |
| **9** | Data structure manipulation | Understand and build custom data structure manipulations | Program to Implement Arrays. | Program to Implement Arrays. |
| **10** | Inheritance | Understand multiple types of inheritance | Program to implement Multiple Inheritance | Program to implement Multiple Inheritance |
| **11** | This pointer | Understand Pointers | Program To Implement This Pointer | Program To Implement This Pointer |
| **12** | Friend class | Understand friend class | Program to Implement Friend class | Program to Implement Friend class |
| **13** | Functional overloading | Understand functional overloading | Program to Implement Function overloading | Program to Implement Function overloading |
| **14** | Operator overloading | Understand operator overloading | Program to Implement Operator Overloading | Program to Implement Operator Overloading |
| **15** | String manipulation | Understand string manipulation | Program to Implement String concepts | Program to Implement String concepts |

**Notes:**

* **Each session** (4 hours) can be structured as:
  + **1 hour** – Concept briefing & demo
  + **2 hours** – Hands-on implementation
  + **1 hour** – Review, viva, and discussion
* **Assessment**: Conduct mini-viva, peer review, or evaluation rubrics weekly.
* Encourage students to **document their code** and maintain a **practical record**.

**Signature of the Course Teacher**

**Signature of the Head of the Department**