NOTICE

Lab Evaluation-2

Course: Object Oriented Programming (UTA018) | III Semester | 2026-2027ODD

Schedule

Date: 27 - 31 October 2026

Time: During regular lab timings

Venue: As per the timetable (The lab evaluations scheduled for Saturday, 31 October 2026, will be conducted for those subgroups whose regular lab sessions are scheduled on Mondays.)

Platform: Code chef in proctored mode

Syllabus: Unit-01, Unit-02, Unit-03 and Unit-04 as per the syllabus. Detailed syllabus is attached at the end.

Important Instructions

- **Remember your CodeChef account password.** Password resetting will not be permitted during the evaluation.
- Report to the laboratory on time for your scheduled evaluation. No extra time will be granted beyond the allotted examination duration.
- Late arrival, password recovery, login difficulties, or other non-genuine delays will not entitle you to additional time.
- The examination must be completed within the scheduled time.
- **No belongings will be allowed inside the contest venue.** This includes mobile phones, water bottles, gadgets, notebooks, stationery, bags, etc. You will be allowed empty hands.
- **The Contest will primarily assess your logical thinking, analytical ability, programming skills, and problem-solving ability.**
- **Zero marks will be awarded if any mischievous, unfair, or inappropriate activity is detected during the evaluation.**
- **No makeup will be conducted under any circumstances.**
- **Do not restrict your preparation to the slides alone.** You are encouraged to practice and explore problems from online programming platforms such as LeetCode, CodeChef, and HackerRank. The slides are indicative and are intended to provide direction for your preparation.

- **Reporting Question-Related Issues**
  - If you identify an issue with a question, inform the invigilator immediately during the examination.
  - Clearly explain the issue and provide any relevant details requested by the invigilator.
  - The invigilator may record the details and capture a screenshot for documentation.
  - Continue attempting the question wherever possible.
  - The evaluation team will review reported issues after the examination. Marks will be awarded if the issue is found to be genuine.
  - Reporting an issue does not entitle you to additional examination time.

Focus on understanding concepts, applying them to problems, debugging your approach, and developing solutions independently.

All the best!

Syllabus

Objects and Classes: Structure in C and C++, Class specification, Objects, Namespaces, Overview of pillars of OOPS, scope resolution, dot, arrow, sizeof operator(without polymorphism), COUT and CIN as object of class ostream and istream (without << and >>).

Inline functions vs Macro substitution (System level), Passing objects as arguments, Returning object from a function, Array of objects, Static keyword with data member, member function and object, Friend function, and Friend classes, Pointer to objects, this pointer, Dynamic Initialization, Dynamic memory allocation, reference variable and its use in functions, pointers, memory allocation of static, pointers/references, const data members, member functions, function with default arguments.

Constructors and its types, Constructor Overloading, Constructors in array of objects, Constructors with default arguments, Dynamic Constructor, Destructor, _const_ keyword with data member, member function and object.

Introduction to Inheritance, Forms of Inheritance (Single, Multiple, Multilevel, Hierarchical and Hybrid) with various modes (Public, Private and Protected), Inheritance with Constructor and Destructor, Benefits and Limitations of Inheritance.

Classification of Polymorphism (Compile-time and Run-time), Compile Time-Function Overloading, Operator Overloading (Unary operator and Binary operator with member function and friend function), Data Conversion (Basic to user-defined, user-defined to basic, one user-defined to another user-defined). Run-time- Pointers to derived class object, Overriding member function, Virtual functions, pure virtual functions, Abstract class. Case Study on Developing an Eco-Friendly Smart Transportation System.