II Sem / N Batch OOP Lab 3rd Feb 2026

Design and implement a **C++ program** to simulate a **Course Admission Application System** for a college.

The system should allow applicants to submit applications for different courses and enable the admission office to review and display applicant details.

System Requirements

1. **Class Design**
   - Create a class named `AdmissionApplication` to store and manage applicant details.
2. **Applicant Data**
   Each application should store the following information:
   - Applicant Name
   - Application ID
   - Course Name
   - Marks obtained
   - Category (FULLTIME / PARTTIME)
   - Admission Status (Applied / Eligible / Not Eligible)
3. **Objects and Object Interaction**
   - Create multiple objects of the class to represent different applicants.
   - Pass **objects as arguments** to functions that evaluate eligibility or compare applicants.
4. **Default Arguments**
   - Use **default arguments** in a function to assign a default category (e.g., "FULLTIME") if not provided by the applicant.
5. **Function Overloading**
   - Overload a function named `calculateEligibility()`:
     - One version checks eligibility based only on marks.
     - Another version checks eligibility based on marks and category(Full time - marks >=90; Part time – marks >=75).
6. **Static Data Members**
   - Use a **static data member** to keep track of the **total number of admission applications received**.
   - Display this count whenever an application is submitted by a static function
7. **Friend Function**
   - Implement a **friend function** that can access private data members of the class to:
     - Compare two applicants and display who has higher marks.
8. **Member Functions**
   - Input applicant details.
   - Display individual application details.
   - Update admission status based on eligibility rules.
9. **Program Execution**
   - The program should be menu-driven, allowing the user to:
     - Submit a new application
     - Display application details
     - Check eligibility
     - View total number of applications received

- The Test Routine must demonstrate the use of:
  - Class and Objects
  - Default arguments
  - Function overloading
  - Passing objects as arguments
  - Static data members
  - Friend functions