**Register Number:**

**Department of Information Technology : MIT Campus : Anna University**

**IT23304 Object Oriented Programming**

**Internal Lab Assessment**

**Class: B.Tech.(I.T)-3/8 Duration : 1 Hr.**

**Date : 19.09.2025 Marks : 25**

Design and implement a C++ program for an **E-Wallet System** using a class EWallet.

**Class: EWallet**

**Attributes**

* string name → User’s name
* int age → User’s age
* string address → User’s address
* double amount → Wallet balance
* static double totalTransacted → Shared among all wallets to track the total amount added/deducted
* const double MIN\_BALANCE → A constant attribute to represent the minimum balance required in the wallet (e.g., ₹100).

**Member Functions**

1. **Constructor**
   * To initialize user details and wallet balance.
2. **Function Overloading**
   * void addMoney(double amt) → Add money to the wallet.
   * void addMoney(double amt, string source) → Add money with a description of the source (e.g., “Bank Transfer”, “UPI”).
3. **Operator Overloading**
   * operator-(double) → Deduct amount from the wallet (ensuring the balance never goes below MIN\_BALANCE).
   * operator+(double) → Add funds to the wallet.
   * friend istream& operator>> → Input user details.
   * friend ostream& operator<< → Display user details and wallet balance.
4. **Destructor**
   * Display the total transacted amount at the end of all transactions.

**Register Number:**

**Department of Information Technology : MIT Campus : Anna University**

**IT23304 Object Oriented Programming**

**Internal Lab Assessment**

**Class: B.Tech.(I.T)-3/8 Duration : 1 Hr.**

**Date : 19.09.2025 Marks : 25**

Design and implement a C++ program for an **Online Shopping Cart System** using a class ShoppingCart.

### ****Class: ShoppingCart****

**Attributes**

* string customerName → Name of the customer
* const int customerID → Unique ID of the customer (must remain constant once initialized)
* double totalAmount → Total bill amount
* static double totalTransactions → Shared among all carts to track the total purchase/return amounts

**Member Functions**

1. **Constructors**
   * Default constructor → Initializes values with defaults.
   * Parameterized constructor → Initializes cart with customer details and initial bill amount.
2. **Function Overloading**
   * void addItem(string itemName, double price) → Add an item with only price.
   * void addItem(string itemName, int quantity, double price) → Add an item with quantity and price.
3. **Operator Overloading**
   * operator-(double) → Apply discount (deduct from bill).
   * operator+(double) → Add delivery charge (increase bill).
   * friend istream& operator>> → Input customer details.
   * friend ostream& operator<< → Display cart details and total amount.
4. **Destructor**
   * Display the total transaction amount processed across all carts at the end of the program.

**Register Number:**

**Department of Information Technology : MIT Campus : Anna University**

**IT23304 Object Oriented Programming**

**Internal Lab Assessment**

**Class: B.Tech.(I.T)-3/8 Duration : 1 Hr.**

**Date : 19.09.2025 Marks : 25**

Design and implement a C++ program for a **Library Management System** using a class LibraryBook.

**Class: LibraryBook**

**Attributes**

* string title → Title of the book
* string author → Author of the book
* int bookID → Unique ID of the book
* bool isIssued → Status of the book (issued or not)
* static int totalBooks → Shared among all books to track the total number of books in the library
* const double FINE\_RATE → A constant fine rate (e.g., ₹2 per day)

**Member Functions**

1. **Constructors**
   * Default constructor → Initializes attributes with default values.
   * Parameterized constructor → Initializes with given book details and updates totalBooks.
2. **Function Overloading**
   * void issueBook() → Issue the book without a due date.
   * void issueBook(int days) → Issue the book with a due date, applying fine if delayed.
3. **Operator Overloading**
   * operator+(int) → Add multiple copies of the same book (increase totalBooks).
   * operator-(int) → Remove copies of a book (decrease totalBooks).
   * friend istream& operator>> → Input book details.
   * friend ostream& operator<< → Display book details.
4. **Destructor**
   * Display the total number of books remaining in the library when the program ends.

**Register Number:**

**Department of Information Technology : MIT Campus : Anna University**

**IT23304 Object Oriented Programming**

**Internal Lab Assessment**

**Class: B.Tech.(I.T)-3/8 Duration : 1 Hr.**

**Date : 19.09.2025 Marks : 25**

Design and implement a C++ program for a **Bank Account Management System** using a class BankAccount.

**Class: BankAccount**

**Attributes**

* string accountHolder → Name of the account holder
* int accountNumber → Unique account number
* double balance → Current account balance
* static double totalDeposits → Shared among all accounts to track the total deposits made
* const double MIN\_BALANCE → A constant value to represent the minimum required balance (e.g., ₹500)

**Member Functions**

1. **Constructors**
   * Default constructor → Initializes attributes with default values.
   * Parameterized constructor → Initializes account with holder name, number, and opening balance.
2. **Function Overloading**
   * void deposit(double amount) → Deposit money into the account.
   * void deposit(double amount, string source) → Deposit money with a source description (e.g., “Cash Deposit”, “Cheque”).
3. **Operator Overloading**
   * operator-(double) → Withdraw money (ensuring balance never goes below MIN\_BALANCE).
   * operator+(double) → Add interest to the account balance.
   * friend istream& operator>> → Input account details.
   * friend ostream& operator<< → Display account details and current balance.
4. **Destructor**
   * Display the total deposited amount across all accounts at the end of the program.

**Register Number:**

**Department of Information Technology : MIT Campus : Anna University**

**IT23304 Object Oriented Programming**

**Internal Lab Assessment**

**Class: B.Tech.(I.T)-3/8 Duration : 1 Hr.**

**Date : 19.09.2025 Marks : 25**

Design and implement a C++ program for a **Student Result Management System** using a class Student.

### ****Class: Student****

**Attributes**

* const int rollNo → Unique, constant roll number of the student (cannot be changed once initialized).
* string name → Student’s name.
* int age → Student’s age.
* double marks → Marks obtained by the student.
* static int studentCount → Shared among all objects to count the total number of students created.
* static double totalMarks → Shared among all objects to track total marks scored by all students.

**Member Functions**

1. **Constructors**
   * Default constructor → Initialize values with defaults.
   * Parameterized constructor → Initialize with given roll number, name, age, and marks.
2. **Function Overloading**
   * void updateMarks(double newMarks) → Update marks directly.
   * void updateMarks(double internalMarks, double externalMarks) → Update marks using internal + external.
3. **Operator Overloading**
   * operator+(double) → Add grace marks to the student.
   * operator-(double) → Deduct penalty marks from the student.
   * friend istream& operator>> → Input student details.
   * friend ostream& operator<< → Display student details.
4. **Static Member Functions**
   * static void showTotalStats() → Display total number of students and total marks scored.
5. **Destructor**
   * Display a message when a student object is destroyed, showing the student’s roll number and marks.