Friend Functions, Constructors, Destructors, Static Variables, and Access Specifiers
Practice set-5

1 Friend Functions

1. Write a C++ program to demonstrate a **friend function** that accesses private members of two different classes.
2. Implement a friend function that allows **two classes to share a common function** to add private variables.
3. Create a class `Rectangle` with private length and width and use a friend function to calculate the **area**.
4. Write a program where a **friend function swaps private variables** of two classes.
5. Implement a **friend function to compare two objects** of a class based on private data.

2 Constructors and Constructor Overloading

1. Write a program with a **default constructor** that initializes values and a function to display them.
2. Create a class `Student` with **parameterized constructors** to initialize different sets of variables.
3. Demonstrate **constructor overloading** with three constructors:
   - One without parameters
   - One with one parameter
   - One with two parameters
4. Implement a **copy constructor** to initialize an object using another object of the same class.
5. Write a class `Complex` to demonstrate constructor overloading by initializing real and imaginary parts in different ways.

3 Destructors

1. Write a program that prints a message when an object is **created and destroyed** using a constructor and destructor.
2. Implement **dynamic memory allocation** in a constructor and deallocate it using a destructor.
3. Write a class `FileHandler` where a destructor automatically **closes a file** when the object goes out of scope.
4. Demonstrate how **destructors** are called when objects are created inside a block.

4 Static Variables and Static Functions

1. Write a class that **counts the number of objects created** using a static variable inside a constructor.
2. Implement a **static function** that accesses a static member variable to keep track of the number of function calls.
3. Create a class `BankAccount` where static variables store the **interest rate** applicable to all accounts.
4. Show how **static variables retain values** between function calls.

5 Public and Private Access Specifiers

1. Write a program to **demonstrate private data members** with getter and setter functions.
2. Implement a **class with a mix of public and private members** and show how functions access them correctly.