|  |  |  |  |
| --- | --- | --- | --- |
| **Object Oriented Programming Lab 04** | | | |
| **Course Code:** | COMP-112L | **Class** | CS (B) |
| **Lab Engineer** | Laiba Khalid | **Semester** | 2nd |
| **Lab Title** | C++ Classes | **Section** | - |
| **Name** |  | **Reg no.** |  |
| **Content Covered** | C++ Class Constructor | | |
| **Instructions:**  • Submit the file with your names following your registration numbers like AI001\_Name.  • Submit soft copy of the report before deadline. Marks will be deducted for late submissions. | | | |

**Lab Tasks**

**Task 1: Default and Parameterized Constructor**

* Create a class Rectangle with attributes length and width.
* Implement:
  + A **default constructor** that sets length and width to 1.
  + A **parameterized constructor** to initialize the attributes.
* Add a method area() to calculate and return the area of the rectangle.
* Create objects using both constructors and display the results.

**Task 2: Constructor Overloading**

* Create a class BankAccount with attributes accountNumber and balance.
* Implement:
  + A **default constructor** that sets balance to 0.0.
  + A **parameterized constructor** to initialize accountNumber and balance.
* Add a method display() to show account details.
* Test constructor overloading by creating multiple objects with different constructors.

**Task 3: Dynamic Memory Allocation in Constructor**

* Create a class Person with:
  + A dynamically allocated char\* name
  + An integer age
* Implement a constructor that:
  + Allocates memory for name
  + Initializes age
* Implement a **destructor** to free allocated memory.
* Create objects and display the details.

**Task 4: Copy Constructor**

* Create a class Student with attributes:
  + name (string)
  + marks (float)
* Implement a **copy constructor** to copy the values of one object into another.
* Create a Student object, copy it to another, and display both.

**Task 5: Constructor with Inheritance**

* Create a base class Vehicle with an attribute brand and a constructor that initializes it.
* Create a derived class Car that adds:
  + model (string)
  + year (int)
* Use the **base class constructor** inside the derived class constructor.
* Create an object of Car and display details.