# Practical No 4

## Aim : Write A Program To Implement Default And Parameterized Constructor

### A. Default Constructor

This C++ code defines a `Student` class with a **default constructor** (a constructor with no parameters) that prompts the user to input values for the object's properties (name, stream, age, roll number). The `display` method is then used to print these values. In the `main` function, a `Student` object is created, automatically calling the default constructor to input values, and then the `display` method is called to show the inputted values.

### B. Parameterized Constructor

This C++ code defines a `Circle` class with a **parameterized constructor** that takes a radius (`double r`) as an argument. This constructor initializes the `Radius` member variable with the provided value.

In other words, when a `Circle` object is created, the constructor is called with a specific radius value, which is then used to calculate and display the circle's circumference and area.

For example, in the `main` function, `Circle c(r);` creates a `Circle` object with the radius value entered by the user, which is then used to calculate and display the circle's properties.

# Practical No 5

## Aim : Write A Program To Implement Single And Multi-Level Inheritance

### A. Single

This C++ code demonstrates single inheritance.

The `Salary` class has a protected field `salary` with a value of 60000.

The `Programmer` class inherits from `Salary` and adds its own fields `Name` and `bonous`. The `display` method in `Programmer` prints out the programmer's name, bonus, salary, and gross salary (salary + bonus).

In the `main` function, a `Programmer` object is created and its `display` method is called, printing out the programmer's details.

### B. Multi-Level

This C++ code calculates and displays the total marks and percentage of a student's grades in four subjects: Physics, Chemistry, Maths, and Biology. It uses multi-level inheritance to organize the data and calculations into three classes: `Data` (stores subject marks), `Sum` (calculates total marks), and `Percent` (calculates percentage). The `main` function creates an instance of `Percent`, prompts the user to input marks, and then displays the results.

# Practical No 6

## Aim : Write A Program To Implement Multiple Inheritance

This C++ code demonstrates multiple inheritance, where the `Output` class inherits from both `Triangle` and `Rectangle`, which inherit from a common base class `Polygon`. This allows the `Output` class to access the common properties (`height` and `width`) from `Polygon` through both `Triangle` and `Rectangle`. The `Output` class then uses scope resolution operators to access the specific `area` method implementations from either `Triangle` or `Rectangle`, depending on user input. This enables the `Output` class to calculate and display the area of either shape.