

# School of Computing and Data Science

## Sai University

### Practice Set 5: C++ Basics

1. Define a class `Rectangle` with data members `length` and `breadth`. Write a constructor to initialize both and a member function to calculate the area.
2. Write a program to create a class `Student` with data members `name`, `age`, and `marks`. Implement both a default constructor and a parameterized constructor to initialize the data.
3. Create a class `Complex` with two data members `real` and `imag`. Overload constructors to:
  - Initialize both to zero (default constructor).
  - Initialize with one value (real part only).
  - Initialize with two values (real and imaginary parts).
4. Implement a class `BankAccount` with data members `accountNumber`, `balance`. Write multiple constructors for:
  - Default initialization.
  - Initialization with account number only.
  - Initialization with account number and balance.
5. Write a C++ program with a class `Time` having data members `hours`, `minutes`, `seconds`. Overload constructors to:
  - Initialize all with zero.
  - Initialize hours and minutes, seconds defaulted to zero.
  - Initialize all three.
6. Design a class `Book` with attributes `title`, `author`, and `price`. Provide constructor overloading to allow creating:
  - Book with title only.
  - Book with title and author.
  - Book with title, author, and price.
7. Create a class `Distance` with data members `feet` and `inches`. Implement constructors for:
  - Default initialization.
  - Initialization using feet only.
  - Initialization using feet and inches.Also, write a function to display the distance.
8. Define a class `Car` with members `brand`, `model`, and `price`. Overload

constructors to:

- Initialize brand only.
- Initialize brand and model.
- Initialize brand, model, and price.

9. Write a program to create a class **Employee** with data members **name**, **id**, and **salary**. Implement:

- A constructor that initializes name and id only.
- A constructor that initializes all three values.

10. Create a class **Fraction** with numerator and denominator. Overload constructors to:

- Initialize fraction as 0/1.
- Initialize with numerator only (denominator = 1).
- Initialize with numerator and denominator.

Also, add a member function to reduce the fraction to simplest form.