SCS 1310 - Object Oriented Programming

Practical 2

Data Types, Structures, and Classes

Student Information Management System

You are hired as a junior C++ developer for a university's IT department. The department needs a simple console-based Student Information Management System. Before you develop the full system, your team lead wants you to complete several foundational tasks related to **data types**, **structures**, and **classes**. These tasks are essential for building the system's core functionality.

Part A – Understanding Data Types

Task A1 – Identify the Data Types

From the scenario, choose the most appropriate data type for the following pieces of student information:

1. Student age → ___________
2. Student first name → ___________
3. Student GPA → ___________
4. Student registration number → ___________
5. Is student active? (Yes/No) → ___________
6. Total marks for a course (0–100) → ___________

Task A2 – Declare Variables

Declare C++ variables for the information above.

Example:

```
int age;
```

Part B – Derived Data Types

Task B1 – Create an Array

Create an array to store the GPAs of 5 students.

Task B2 – Create a Pointer

Create a pointer to your `age` variable and display its memory address.

Task B3 – Function Practice

Write a function called `printWelcome()` that prints: *"Welcome to the Student Information System"*.

Part C – Structures

Guiding Context:

The first module of your system needs to store basic details of a student.

Task C1 – Define a Structure

Create a structure named **Student** containing:

- name (char array)
- age (int)
- height (float)
- weight (float)

Task C2 – Declare a Structure Variable

Declare a variable `std1` of type `Student` and manually assign values.

Task C3 – Print Structure Data

Write code to print the values of `std1`.

Part D – Classes and Objects

Guiding Context:

The department wants to upgrade the system using OOP concepts.

Task D1 – Create a Class

Create a class named **Date** with:

- **Private data members:** day, month, year
- **Public member functions:**
  - `setDate(int d, string m, int y)`
  - `getDate()` – prints the date

Task D2 – Create an Object

Create an object named **today**. Use `setDate()` to assign:

- 13 / November / 2025

Call `getDate()` to print it.

Task D3 – Use a Pointer to an Object

Create a pointer `ptrDate` and assign it the address of `today`. Use the `->` operator to display the date.

Part E – Class Extension

Guiding Context:

Your supervisor now wants the system to store registration details.

Task E1 – Create a Class Named `StudentReg`

Include:

- **Public data members:** regNo, name, degree
- **A function:** `displayDetails()`

Task E2 – Create Multiple Objects

Create an array of 3 `StudentReg` objects.

Input values manually and call `displayDetails()` for each.