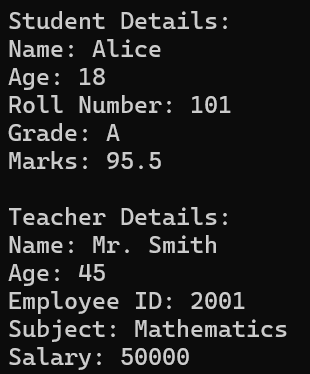
1. Write a C++ program for managing information about people in a school system. The base class should be named ***Person***, containing attributes for name and age.

Derive two classes from ***Person***: ***Student*** and ***Teacher***.

The ***Student*** class should include additional attributes for rollNumber, grade, and marks, while the ***Teacher*** class should have attributes for employeeID, subject, and salary.

Each class should have a constructor to initialize the object with appropriate values and a member function *displayDetails()* to print all the information about the person, including the specific attributes for students and teachers. Use inheritance by creating objects of child classes, setting their attributes, and displaying using the *displayDetails()* function.

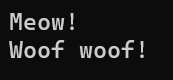
**Sample Run:**



1. Write a C++ program to implement a class hierarchy using polymorphism between three classes, ***Animal***, ***Cat***, and ***Dog***. The ***Animal*** class is the base class and ***Cat*** & ***Dog*** are two derived classes. In the program, derived classes should override the base class's virtual function *makeSound()* to produce their specific sounds.

In the main function, objects of the derived classes are managed with pointers to the base class, demonstrating polymorphism, where the correct functions are called based on the object type. Consider the sample run given below.

**Sample Run:**



1. Write a C++ program to manage employees in a company. Your program should represent different types of employees and their roles, where each employee type has a different description of their role in the company. The roles of these employees are as follows:

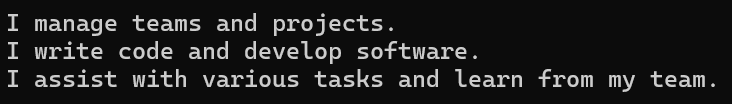
Manager; manages teams and projects, Developer; writes code and develops software, Intern; assists with various tasks and learns from the team.

Create a base class ***Employee*** that has a function *describeRole()* that prints “I am a generic employee.” Override this function in derived classes Manager, Developer, and Intern. Give the messages respectively, “I manage teams and projects.”, “I write code and develop software.”, “I assist with various tasks and learn from my team.” (Use **polymorphism**.)

In the main function,

Create objects of Manager, Developer, and Intern classes. Create an array of pointers to the Employee class. Store the addresses of the Manager, Developer, and Intern objects in this array. Construct a loop through the array and call the describeRole() function for each employee using polymorphism. Then, print the correct role of the employee.

**Sample Run:**



1. A ***Shape*** abstract class is defined, providing an interface for calculating the areas of various geometric shapes. The *Rectangle* and *Circle* classes are derived from this abstract class and implement the calculation of the area method. Consider the following:

* The *Shape* class must contain a pure virtual function named *calculateArea()*.
* The *Rectangle* class has the attributes width and height, and implements the *calculateArea()* function.
* The ***Circle*** class has the attribute radius and implements the *calculateArea()* function.

In the main function, create a ***Rectangle*** and ***Circle*** object to calculate the areas of these shapes.

**Sample Run:**

