Module -3 introduction To OOPS Programming

Lab Exercise

❖Introduction To C++:-

→1. First C++ Program: Hello World

o Write a simple C++ program to display "Hello, World!".

Objective: Understand the basic structure of a C++ program, including #include, main(), and cout.

Ans.

```
#include<iostream>
using namespace std;
int main(){
    cout<<"Hello World"<<endl;
    return 0;
}
```

→2. Basic Input/Output

o Write a C++ program that accepts user input for their name and age and then displays a personalized greeting.

Objective: Practice input/output operations using cin and cout.

Ans.

```
#include<iostream>
```

```
using namespace std;
int main(){
	int age;
	char name[100];
	cout<<" Enter Your Name= ";
	cin>>name;
	cout<<" Enter Your Age= ";
	cin>>age;
	cout<<" Hi,"<<name<<" Your Current Age is "<<age<<endl;
	return 0;
}
```

→3. POP vs. OOP Comparison Program

Write two small programs: one using Procedural Programming (POP) to calculate the area of a rectangle, and another using Object-Oriented Programming (OOP) with a class and object for the same task.

Objective: Highlight the difference between POP and OOP approaches.

Ans.

→In Pop(Procedural Oriented Programming):-

```
#include<stdio.h>
```

```
int main(){
	int length,width,area;
	printf("\n Enter the Length is= ");
	scanf("%d",&length);
	printf("\n Enter the Width is= ");
	scanf("%d",&width);
	area=length*width;

	printf("\n Length is =%d",length);
	printf("\n width is =%d",width);
	printf("\n Area is =%d",area);
	return 0;
}
```

→In OOP(Object Oriented Programming):

```
#include<iostream>
using namespace std;
class area{
	int length,width,area;
	public:
		void getdata(){
			cout<<"Enter the Length= ";
```

```
            cin>>length;
            cout<<"Enter the width= ";
            cin>>width;
        }
        void display(){
            area=length*width;
            cout<<"Length= "<<length<<endl<<"Width=
"<<width<<endl<<"Area= "<<area<<endl;
        }
};
int main(){
    area a;
    a.getdata();
    a.display();
    return 0;
}
```

→4.Setting Up Development Environment

o Write a program that asks for two numbers and displays their sum. Ensure this is done after setting up the IDE (like Dev C++ or CodeBlocks).

o **Objective**: Help students understand how to install, configure, and run programs in an IDE.

Ans.

```
#include<iostream>
using namespace std;
int main(){
    int num1,num2,sum;
    cout<<"Enter The Number1= and  Number2= ";
    cin>>num1>>num2;
    sum=num1+num2;
    cout<<"Sum of "<<num1<<" And "<<num2<<" is = "<<sum;
    return 0;
}
```

❖**Variable ,Data Type And Operator:**-

→1. Variables and Constants

o Write a C++ program that demonstrates the use of variables and constants. Create variables of different data types and perform operations on them.

o **Objective**: Understand the difference between variables and constants

→Ans.

```
#include<iostream>
using namespace std;
int main(){
	int num1,num2;
	cout<<"Enter the num1= ";
	cin>>num1;
	cout<<"Enter the num2= ";
	cin>>num2;
	bool temp=true;
	while(temp){
		char choice;
		cout<<"'+'. Addition "<<endl;
		cout<<"'-'. Subtraction "<<endl;
		cout<<"'*'. Multiplication"<<endl;
		cout<<"'/'. Division "<<endl;

		cout<<"'E'. Exit "<<endl;

		cout<<"Enter your Choice =";
		cin>>choice;
		switch(choice){
```

```
                case '+':cout<<"Addition is=
"<<num1+num2<<endl;
                    break;
                case '-':cout<<"Subtraction is= "<<num1-
num2<<endl;
                    break;
                case '*':cout<<"Multiplication is=
"<<num1*num2<<endl;
                    break;
                case '/':
                   float ans;
                   ans=(float)num1/num2;
                   cout<<"Division is= "<<ans<<endl;
                    break;

                case 'E':
                      temp=false;
                    break;
                default:cout<<"Enter Valid Input "<<endl;

            }
        }
```

```
    return 0;
}
```

→2.Type Conversion

o Write a C++ program that performs both implicit and explicit type conversions and prints the results.

o Objective: Practice type casting in C++

Ans:-

```
#include<iostream>
using namespace std;
int main(){
    //implicit Type convertion
    int a=10;
    float b=a;

    cout<<"Implicit convertion "<<endl;
    cout<<"Integer value= "<<a<<endl;
    cout<<" converted into float = "<<b<<endl;

    float f=3.14;
    int i=(int)f;
```

```
    cout<<"Explicit Convertion "<<endl;
    cout<<" Float Value= "<<f<<endl;
    cout<<"converted in int = "<<i<<endl;

    double d=9.99;
    int n=static_cast<int>(d);
    cout<<"use  static cast "<<endl;
    cout<<"Double value= "<<d<<endl;
    cout<<"converted into int ="<<n<<endl;

    return 0;
}
```

→3. Operator Demonstration

o Write a C++ program that demonstrates arithmetic, relational, logical, and bitwise operators. Perform operations using each type of operator and display the results.

o Objective: Reinforce understanding of different types of operators in C++.

Ans.

```
#include<iostream>
```

```
#include<string>
using namespace std;
int main(){
	int num1,num2;
	bool temp=true;
	while(temp){

		cout<<"A. Arithmetic Operator"<<endl;
		cout<<"B. Bitwise Operator"<<endl;
		cout<<"R. Relational Operator"<<endl;
		cout<<"L. Logical Operator"<<endl;
		cout<<"E. Exit"<<endl;

		char choice;
		cout<<"Enter Your Choice= ";
		cin>>choice;
		switch(choice){
			case 'A':

				cout<<"A. Addition"<<endl;
				cout<<"S. Subtraction"<<endl;
```

```
                cout<<"M. Multiplication"<<endl;
                cout<<"D. division"<<endl;
                cout<<"E. Exit"<<endl;
                char ch1;
                cout<<"Enter Your Choice= ";
                cin>>ch1;


                switch(ch1){
                      case 'A':
                            cout<<"Enter the num1 and num2= ";
                            cin>>num1>>num2;
                            cout<<"Addition is=
"<<num1+num2<<endl;
                         break;
                      case 'S':

                            cout<<"Enter the num1 and num2= ";
                            cin>>num1>>num2;
                            cout<<"subtraction is= "<<num1-
num2<<endl;
                         break;
```

```
                    case 'M':

                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<"multiplication is=
"<<num1*num2<<endl;
                      break;
                    case 'D':

                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        float ans;
                        ans=(float)num1/num2;
                        cout<<"Division is= "<<ans<<endl;
                      break;

                    Default:
                        cout<<"Enter Valid Input"<<endl;
                }

                break;
```

```
            case 'B':
                cout<<"1. And(&) operator"<<endl;
                cout<<"2. Or(|) operator"<<endl;
                cout<<"3. Eor(^) operator"<<endl;
                cout<<"4. not(~) operator"<<endl;
                cout<<"5. right shift(>>) operator"<<endl;
                cout<<"6. left shift(>>) operator"<<endl;
                int n;
                cout<<"Enter your choice= ";
                cin>>n;
                switch(n){
                    case 1:
                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<" & "<<num2<<"is =
"<<(num1&num2)<<endl;
                        break;
                    case 2:

                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
```

```
                cout<<num1<<" | "<<num2<<"is =
"<<(num1|num2)<<endl;
                break;
            case 3:

                cout<<"Enter the num1 and num2= ";
                cin>>num1>>num2;
                cout<<num1<<" ^ "<<num2<<"is =
"<<(num1^num2)<<endl;
                break;
            case 4:

                cout<<"Enter the num1 = ";
                cin>>num1;
                cout<<"(~"<<num1<<") =
"<<(~num1)<<endl;
                break;
            case 5:
                cout<<"Enter the num = ";
                cin>>num1;
                cout<<num1<<">>1  =
"<<(num1>>1)<<endl;
```

```
                    break;
                case 6:
                    cout<<"Enter the num = ";
                    cin>>num1;
                    cout<<num1<<"<<1  =
"<<(num1<<1)<<endl;
                    break;
                }

            break;
            case 'R':

                cout<<"g. Greater than"<<endl;
                cout<<"l. Less than"<<endl;
                cout<<"n. Greater than Equal to "<<endl;
                cout<<"m. Less than equal to"<<endl;
                cout<<"e. Equal to "<<endl;
                cout<<"r. Not Equal to"<<endl;
                cout<<"E. Exit"<<endl;
                char ch2;
                cout<<"Enter Your Choice= "<<endl;
                cin>>ch2;
```

```
                switch(ch2){
                    case 'g':

                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<"> "<<num2<<" is =
"<<(num1>num2)<<endl;;
                        break;
                    case 'l':

                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<" < "<<num2<<" is =
"<<(num1<num2)<<endl;;
                        break;
                    case 'n':

                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<" >= "<<num2<<" is
="<<(num1>=num2)<<endl;;
                        break;
```

```
                    case 'm':


                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<" <= "<<num2<<" is
="<<(num1<=num2)<<endl;;
                        break;
                    case 'e':


                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<" == "<<num2<<" is
="<<(num1==num2)<<endl;;
                        break;
                    case 'r':


                        cout<<"Enter the num1 and num2= ";
                        cin>>num1>>num2;
                        cout<<num1<<" != "<<num2<<" is
="<<(num1!=num2)<<endl;;
                        break;
                    default:
```

```
                    cout<<"Enter Valid Detail"<<endl;

                }
            break;
            case 'L':

                int c1,c2;
                 cout<<"A. Logical and"<<endl;
                 cout<<"O. Logical Or"<<endl;
                 cout<<"N. Logical not"<<endl;
                 cout<<"E. Exit"<<endl;
                 char ch3;
                 cout<<"Enter Your Choice= ";
                 cin>>ch3;
                 switch(ch3){
                     case 'A':

                     cout<<"Enter the C1 and c2= ";
                     cin>>c1>>c2;
                         cout<<c1<<" && "<<c2<<" is =
"<<(c1&&c2)<<endl;
                         break;
```

```
                        case 'O':

                        cout<<"Enter the C1 and c2= ";
                        cin>>c1>>c2;
                              cout<<c1<<" || "<<c2<<" is =
"<<(c1||c2)<<endl;
                              break;

                        case 'N':

                        cout<<"Enter the C1= ";
                        cin>>c1;
                              cout<<" ! "<<c1<<" is =
"<<!(c1)<<endl;
                              break;
                        default:cout<<"Enter valid Input"<<endl;

                    }
              break;
              case 'E':
                    temp=false;
```

```
            break;
            default:
              cout<<"Enter Valid Input"<<endl;
        }


    }
    return 0;
}
```

❖Control Flow Statement:-

→1. Grade Calculator

o Write a C++ program that takes a student's marks as input and calculates the grade based on if-else conditions.

o **Objective**: Practice conditional statements (if-else).

Ans.

```
#include<iostream>
using namespace std;
int main(){
    int marks;
    cout<<"Enter the student marks= ";
    cin>>marks;
```

```
        cout<<endl;
        char grade;
        if(marks>80){
              grade='A';
        }
        else if(marks<=80 && marks>=50){
        grade='B';
    }
     else if(marks<50 && marks>=35){
        grade='C';
        }
        else{
              grade='D';
        }
        cout<<"Grade Obtain By Student is = "<<grade<<endl;
        return 0;
}
```

→2. Number Guessing Game

o Write a C++ program that asks the user to guess a number between 1 and 100. The program should provide hints if the

guess is too high or too low. Use loops to allow the user multiple attempts.

o **Objective**: Understand while loops and conditional logic.

Ans.

```
#include<iostream>
using namespace std;
int main(){
	int num=47;
	int guess;
	int count=10;

	while(count!=0){

		cout<<"Total Remaining Attempt= "<<count<<endl;
		cout<<"Guess the number between 1 To 100 =";
	   cin>>guess;
	   cout<<endl;

		if(guess==47){
			cout<<"You Guess  Correct number,You Won The Car."<<endl;
			break;
```

```
            }
            else if(guess>47){
                  cout<<"You Guess To high number"<<endl;
            }
            else{
                  cout<<"You  Guess To low Number"<<endl;
            }
            count--;

      }
      return 0;
}
```

→3. Multiplication Table

o Write a C++ program to display the multiplication table of a given number using a for loop.

o **Objective**: Practice using loops.

Ans.

```
#include<iostream>
using namespace std;
int main(){
      int num,i;
```

```
	cout<<"Enter The Number = ";
	cin>>num;
	cout<<"Multiplication Table Of Given Number "<<num<<" is ="<<endl;
	for(i=1;i<=10;i++){
		cout<<num<<"  "<<i<<"  "<<num*i<<endl;
	}
	return 0;
}
```

→4. Nested Control Structures

o Write a program that prints a right-angled triangle using stars (*) with a nested loop.

o **Objective**: Learn nested control structures.

Ans.

```
#include<iostream>
using namespace std;
int main(){
	int row,i,j;
	cout<<"Enter the Row= ";
	cin>>row;
	for(i=1;i<=row;i++){
```

```
            for(j=1;j<=i;j++){
                  cout<<"* ";
            }
            cout<<endl;
      }
      return 0;
}
```

❖**Function And Scope:**-

→1. Simple Calculator Using Functions

o Write a C++ program that defines functions for basic arithmetic operations (add, subtract, multiply, divide). The main function should call these based on user input.

o **Objective**: Practice defining and using functions in C++.

Ans.

```
#include<iostream>
using namespace std;


void add(int n1,int n2){
      cout<<"Addition Of "<<n1<<" And "<<n2<<" is = "<<n1+n2<<endl;
      cout<<endl<<endl;
```

```
}
void sub(int n1,int n2){
	cout<<"Subtraction Of "<<n1<<" And "<<n2<<" is = "<<n1-n2<<endl;
	cout<<endl<<endl;
}
void mul(int n1,int n2){
	cout<<"Multiplication Of "<<n1<<" And "<<n2<<" is = "<<n1*n2<<endl;
	cout<<endl<<endl;
}
void div(int n1,int n2){
	float ans;
	ans=(float)n1/n2;
	cout<<"Division Of "<<n1<<" And "<<n2<<" is = "<<ans<<endl;
	cout<<endl<<endl;
}

int main(){
	int choice,num1,num2;
	bool temp=true;
```

```
while(temp){

cout<<"1. Addition "<<endl;
cout<<"2. Subtraction "<<endl;
cout<<"3. Multiplication "<<endl;
cout<<"4. Division "<<endl;
cout<<"5. Exit"<<endl;
cout<<endl;

cout<<"Enter Your choice= ";
cin>>choice;
switch(choice){
	case 1:

	cout<<endl;
	cout<<"Enter  The Number1= ";
	cin>>num1;

	cout<<"Enter  The Number2= ";
	cin>>num2;
		add(num1,num2);
```

```
        break;
  case 2:

cout<<endl;
cout<<"Enter  The Number1= ";
cin>>num1;

  cout<<"Enter  The Number2= ";
  cin>>num2;
        sub(num1,num2);
        break;
  case 3:

  cout<<endl;
  cout<<"Enter  The Number1= ";
  cin>>num1;

  cout<<"Enter  The Number2= ";
  cin>>num2;
        mul(num1,num2);
        break;
```

```
            case 4:


            cout<<endl;
            cout<<"Enter  The Number1= ";
            cin>>num1;


            cout<<"Enter  The Number2= ";
            cin>>num2;
                  div(num1,num2);
                  break;
            case 5:
                  temp=false;
                  break;
            default:cout<<"Enter Valid Input"<<endl;
      }
}
      return 0;
}
```

→2. Factorial Calculation Using Recursion

o Write a C++ program that calculates the factorial of a number using recursion.

o **Objective**: Understand recursion in functions.

Ans.

```
#include<iostream>
using namespace std;
int factrial(int n){
	if(n==0||n==1){
		return 1;
	}
	else {
		return n*factrial(n-1);
	}
}
int main(){
	int num;
	cout<<"Enter The number= ";
	cin>>num;
	int res=factrial(num);

	cout<<"factorial of "<<num<<" is = "<<res;

}
```

→3. Variable Scope

o Write a program that demonstrates the difference between local and global variables in C++. Use functions to show scope.

o **Objective**: Reinforce the concept of variable scope.

Ans.

```
#include<iostream>
#include<string>
using namespace std;
int num=1000;//global scope or global variable
int inc_10()
{
	num+=200;

	return 0;
}
int main()
{
	cout<<"Before change in number "<<num<<endl;
	inc_10();
	cout<<"After change in number "<<num<<endl;
```

}

❖Array And String:-

→1. Array Sum and Average

o Write a C++ program that accepts an array of integers, calculates the sum and average, and displays the results.

o **Objective**: Understand basic array manipulation.

Ans.

```
#include<iostream>
using namespace std;
int main(){
    int size,i,sum=0;
    float average;
    cout<<"Enter the size of array= ";
    cin>>size;
    int array[size];
    for(i=0;i<size;i++){
        cout<<"Enter the element in array["<<i<<"] is = ";
```

```
        cin>>array[i];
    }
    for(i=0;i<size;i++){
        cout<<"Array a["<<i<<"] is = "<<array[i]<<endl;
        sum=sum+array[i];
    }
    cout<<"Sum of array element is= "<<sum<<endl;
    cout<<"Average of Array Ellement is = "<<sum/size;
    return 0;
}
```

→2. Matrix Addition

o Write a C++ program to perform matrix addition on two 2x2 matrices.

o **Objective**: Practice multi-dimensional arrays.

Ans.

```
#include<iostream>
using namespace std;
int main(){
    int row,col,i,j;
    cout<<"Enter the Row= ";
    cin>>row;
```

```
cout<<"Enter the Col= ";
cin>>col;
int arr1[row][col],arr2[row][col],add[row][col];
cout<<"Enter Element in Arr1= "<<endl;
for(i=0;i<row;i++){
	for(j=0;j<col;j++){
		cout<<"Enter the Arr1["<<i<<"]["<<j<<"]= ";
		cin>>arr1[i][j];
	}
}
cout<<"Enter Element in Arr2= "<<endl;
for(i=0;i<row;i++){
	for(j=0;j<col;j++){
		cout<<"Enter the Arr2["<<i<<"]["<<j<<"]= ";
		cin>>arr2[i][j];
	}
}
for(i=0;i<row;i++){
	for(j=0;j<col;j++){
		add[i][j]=arr1[i][j]+arr2[i][j];
	}
```

```
}


cout<<"Array 1= "<<endl;
for(i=0;i<row;i++){
	for(j=0;j<col;j++){
		cout<<arr1[i][j]<<"  ";
	}
	cout<<endl;
}
cout<<"Array 2= "<<endl;
for(i=0;i<row;i++){
	for(j=0;j<col;j++){
		cout<<arr2[i][j]<<"  ";
	}
	cout<<endl;
}
cout<<"Addition = "<<endl;
for(i=0;i<row;i++){
	for(j=0;j<col;j++){
		cout<<add[i][j]<<"  ";
	}
```

```
        cout<<endl;
    }
    return 0;
}
```

→3. String Palindrome Check

o Write a C++ program to check if a given string is a palindrome (reads the same forwards and backwards).

o Objective: Practice string operations

Ans.

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string str,reverse;
    cout<<"Enter The string= ";
    cin>>str;
    int i;
    for(i=str.length()-1;i>=0;i--){
        reverse+=str[i];
    }
```

```
    if(str==reverse)
    {
        cout<<"this string is Palindrome string."<<endl;
    }
    else{

        cout<<"this string is not a Palindrome string."<<endl;
    }
    return 0;
}
```

❖ Introduction to Object-Oriented Programming

→1. Class for a Simple Calculator

o Write a C++ program that defines a class Calculator with functions for addition, subtraction, multiplication, and division. Create objects to use these functions.

o Objective: Introduce basic class structure.

Ans.

```
#include<iostream>
using namespace std;
int num1,num2;
class calculator{
      public:
            void add(){

        cout<<"enter Num1 and num2= ";
        cin>>num1>>num2;
        cout<<"Addition is= "<<num1+num2<<endl;
           }
           void sub(){



        cout<<"enter Num1 and num2= ";
        cin>>num1>>num2;
        cout<<"Subtraction is= "<<num1-num2<<endl;
```

```
        }
        void mul(){

      cout<<"enter Num1 and num2= ";
      cin>>num1>>num2;
      cout<<"Multiplication  is= "<<num1*num2<<endl;

        }
        void div(){
        float ans;
      cout<<"enter Num1 and num2= ";
      cin>>num1>>num2;
        ans=(float)num1/num2;
      cout<<"division is= "<<ans<<endl;
        }
};
int main(){
    int ch;
    while(ch!=5){
```

```
cout<<"1. Addition "<<endl;
cout<<"2. Subtraction "<<endl;
cout<<"3. Multiplication "<<endl;
cout<<"4. Division "<<endl;
cout<<"5. Exit"<<endl;
cout<<endl;
cout<<"Enter Your choice ";
cin>>ch;

calculator objcal;

switch(ch){
      case 1:

     objcal.add();
           break;
      case 2:
           objcal.sub();

           break;
      case 3:
```

```
            objcal.mul();


                break;
            case 4:
                objcal.div();


                break;
            case 5:
                break;
            default:
                cout<<"Enter valid choice= "<<endl;
        }
}
        //objcal.add();
//      objcal.sub();
//      objcal.div();
//      objcal.mul();
        return 0;
}
```

→2. Class for Bank Account

o Create a class BankAccount with data members like balance and member functions like deposit and withdraw. Implement encapsulation by keeping the data members private.

o **Objective**: Understand encapsulation in classes.

Ans.

```
#include<iostream>
using namespace std;
int ch;
class BankAccout{
	private:
	  float	balance=1000;
	public:
		/*void setbalance(float bal){
			balance=bal;
		}*/
		void setdepo(float bal){
			balance=balance+bal;
			cout<<"Deposit money successfull"<<endl;
			cout<<"Current balance is =
"<<getbalance()<<endl;

		}
```

```
        void setwith(float bal){
            balance=balance-bal;
            cout<<"Withdraw money successfull"<<endl;
            cout<<"Current balance is =
"<<getbalance()<<endl;

        }
        float getbalance(){
            return balance;
        }
        //void withdraw(){

        //}
        //void balancecheck(){
            //getbalance();
        //}
};
int main(){
    BankAccout obj;
    while(ch!=4){

    cout<<"1. Balance check "<<endl;
```

```
    cout<<"2. Deposit Money "<<endl;
    cout<<"3. Withraw Money "<<endl;
    cout<<"4. Exit "<<endl;
    cout<<endl;
    cout<<"Enter Your Choice = "<<endl;
    cin>>ch;
    switch(ch){
        case 1:
            cout<<"Available Balance is=
"<<obj.getbalance()<<endl;
            break;
        case 2:
            float dep;
            cout<<"enter deposit amount= ";
            cin>>dep;
            obj.setdepo(dep);
            break;
        case 3:
            float with;
            cout<<"Enter Withdraw Amount= ";
            cin>>with;
            if(with<obj.getbalance()){
```

```
            obj.setwith(with);}
            else{
                  cout<<"insufficient balance,please check balance
first "<<endl;
                  }
            break;
      case 4:
            break;
      default:
            cout<<"Enter Valid input "<<endl;

      }
}
      return 0;
}
```

→3. Inheritance Example

o Write a program that implements inheritance using a base class Person and derived classes Student and Teacher. Demonstrate reusability through inheritance.

o **Objective**: Learn the concept of inheritance.

Ans.

```
#include<iostream>
#include<string>
using namespace std;
string name1;
string department;
string subject;
class Person{
	public:
		void name(){
			cout<<"Enter Name= ";
			getline(cin,name1);
		}
		void depart(){
			cout<<"Enter Department Name= ";
			getline(cin,department);
		}
		void sub(){
			cout<<"Enter Subject Name= ";
			getline(cin,subject);
```

```
        }

};
class Student:public Person{
    public:
        void display(){
            cout<<"student name is = "<<name1<<endl;
            cout<<"student department is= "<<department<<endl;
            cout<<"Student Subject is ="<<subject<<endl;

    }

};
class Teacher:public Person{
    public:
    void display(){
            cout<<"Teacher name is = "<<name1<<endl;
            cout<<"Teacher department is= "<<department<<endl;
            cout<<"Teacher Subject is ="<<subject<<endl;
```

```
    }
};
int main(){
    cout<<"Student Information section "<<endl;
    Student obj1;
    obj1.name();
    obj1.depart();
    obj1.sub();
    obj1.display();

    cout<<"Teacher Information section "<<endl;
    Teacher obj2;
    obj2.name();
    obj2.depart();
    obj2.sub();
    obj2.display();
    return 0;
}
```