Assignment - 3

Q.1 no:-1 First C++ Program: Hello World.

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

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

Ans :-

- There are the program of Hello World :-

```
#include<iostream>
int main()
{
    std::cout<<"Hello world";
    return 0;
}
```

Q.1 no.2:- Basic Input/Output.

Ans :-

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

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

- There are the input of program :-

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

int main()
{
    char Name[25];
    int Age;

    cout << "Enter the Name: ";
    cin >> Name;

    cout << "Enter the Age: ";
    cin >> Age;

    cout << "Name = " << Name << endl;
    cout << "Age = " << Age << endl;

    return 0;
}
```

Q.1 no.3:- POP vs. OOP Comparison Program :

o 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.

o Objective: Highlight the difference between POP and OOP approaches.

Ans :-

- This code is POP calculate of area of a rectangle :-

```
#include<stdio.h>
int main()
{
    int length,width,Area;
    printf("\n Enter the length = ");
    scanf("%d",&length);
    printf("\n Enter the width = ");
    scanf("%d",&width);

    Area = length * width;

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

- There are the input of oop :-

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

```
int main()
{
   int length, width, Area;

   cout << "Enter the length: ";
   cin >> length;

   cout << "Enter the width: ";
   cin >> width;

   cout << "Enter the Area: ";
   Area = length * width;

   cout << "Length = " << length << endl;
   cout << "Width = " << width << endl;
   cout << "Area = " << Area;


   return 0;
}
```

Q.1 no.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 :-

- There are the code of two numbers of sum :-

```
// sum of numbers
#include<iostream>
using namespace std;
int main()
{
    int num1,num2,choice;
    cout << "Enter the number = ";
    cin >> num1;
    cout <<"Enter the number = ";
    cin >> num2;


    bool temp=true;
    while(temp)
    {
            cout << "\n1.Addition";
      cout << "\n2.subtrction";
      cout << "\n3.divison";
      cout << "\n4.multiplcation";
      cout << "\n5.exit";
      cout << "\nEnter your choice :";
    cin >> choice;
```

```
        switch(choice)
    {
        case 1:
            cout << num1 + num2;
        break;

        case 2:
            cout << num1 - num2;
        break;

        case 3:
           cout << num1 / num2;
        break;

        case 4:
           cout << num1 * num2;
        break;

        case 5:
            temp=false;
        break;
    }
    }

}
```

Q.2 Variables, Data Types, and Operators.

Q.1 no.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 :-

- There are the code of variables and constants :-

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

int main() {

   const int year = 2025;

   int age = 20;
   float height = 5.9;
   char initial = 'J';


   cout << "Year: " << year << endl;
   cout << "Age: " << age << endl;
   cout << "Height: " << height << " feet" << endl;
   cout << "Initial: " << initial << endl;


   int nextYearAge = age + 1;
   cout << "Age next year: " << nextYearAge << endl;
   return 0;
}
```

- The variable are change any time and constants are not change the value in the program.
- The variable are not declare for value but constants are declare for particular value or function.

Q.2 no.2:- Type Conversion

o Write a C++ program that performs both implicit and explicit type conversions and prints the results. 13 TOPS Technologies 2024

o Objective: Practice type casting in C++.

Ans :-

```
    // implict the value
#include<iostream>
using namespace std;

int main()
{
        int a=10;
        float b = 5.2;
        float result = a + b;

        cout << "result:" << result << endl;
        return 0;
}
```

- Implicit is a automatically convertible type of variable.

- There are the explicit type casting code :-

```
//Explict method program
#include<iostream>
using namespace std;
class employee
{
        int salary;
        public :

void setsalary(int sal)
{
        salary = sal;
}
int getsalary()
{
        return salary;
}
};

int main()
{
   employee p;

   int sal;
   cout << "Enter salary: ";
```

```
    cin >> sal;


    p.setsalary(sal);
    cout << "The salary is: " << p.getsalary() << endl;


    return 0;
}
```

- The explicit method was conversation means manually converting one data type into another program.

Q.2 no.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 :-

- This code are arithmetic operator :-

```
// arithmatic operation
#include<iostream>
#include<string>
using namespace std;
class arith
{

public :
    arith(int a,int b)
    {
        cout <<"1.addition" << endl;
```

```
            cout <<"2.subtraction" << endl;
            cout <<"3.multipition" << endl;
            cout <<"4.divison" << endl;

            int choice;
            cout <<"Enter the choice =";
            cin >> choice;
            switch(choice)
            {
                  case 1:
                        cout <<"addition of = "<< a+b;
                  break;
                  case 2:
                        cout <<"subtraction of =" << a-b;
                  break;
                  case 3:
                     cout <<"multipition of =" << a*b;
                  break;
                  case 4:
                     cout <<"divison of =" << a/b;
                  break;

            }
      }
};

int main ()
{
      arith(14,4);
      return 0;
}
```

- This code are relational operator :-

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

int main()
{
    int a,b;

    cout << "Enter the first number:";
    cin >> a;

    cout << "Enter the second number:";
    cin >> b;

    cout << "\nResult in relational opreation:\n";

    cout <<"a==b:" << (a==b) << endl;
    cout <<"a!=b:" << (a!=b) << endl;
    cout <<"a>=b:" << (a>=b) << endl;
    cout <<"a<=b:" << (a<=b) << endl;
    cout <<"a<b:" << (a<b) << endl;
    cout <<"a>b:" << (a>b) << endl;

    return 0;
}
```

- This is the logical operator code are here :-

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

int main()
{
    int age;
    int marks;
```

```
    cout << "\nEnter the age:";
    cin >> age;

    cout << "\nEnter the mark:";
    cin >> marks;

    // logical AND(&&)
    if(age>18 && marks>50)
    {
        cout << "you are eligable!" << endl;
    }
    else
    {
        cout << "you are not eligable" << endl;
    }

    // logical OR(||)
    if(age>18 || marks>50)
    {
        cout << "you meet the one condition." << endl;
    }

    if(!(age>18))
    {
        cout << "you are NOT older than 18:" << endl;
    }
    return 0;
}
```

- This the code of bitwise operator are here :-

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

void arithmeticOperators(int a, int b) {
   char op;
   bool arithLoop = true;

   while (arithLoop) {
      cout << "\n\n--- Arithmetic Operations ---";
      cout << "\n+ : Addition";
      cout << "\n- : Subtraction";
      cout << "\n* : Multiplication";
      cout << "\n/ : Division";
      cout << "\nE : Exit to Main Menu";
      cout << "\nEnter your operator: ";
      cin >> op;

      switch (op) {
      case '+':
         cout << "Result: " << a + b << "\n";
         break;
      case '-':
         cout << "Result: " << a - b << "\n";
         break;
      case '*':
         cout << "Result: " << a * b << "\n";
         break;
      case '/':
         if (b != 0)
            cout << "Result: " << float(a) / b << "\n";
         else
            cout << "Error: Division by zero!\n";
         break;
      case 'e':
      case 'E':
```

```
            arithLoop = false;
            break;
        default:
            cout << "Invalid operator!\n";
        }
    }
}

void relationalOperators(int a, int b) {
    char op;
    bool relLoop = true;

    while (relLoop) {
        cout << "\n\n--- Relational Operations ---";
        cout << "\na. > (greater than)";
        cout << "\nb. < (less than)";
        cout << "\nc. >= (greater than or equal)";
        cout << "\nd. <= (less than or equal)";
        cout << "\ne. == (equal)";
        cout << "\nf. != (not equal)";
        cout << "\ng. Exit to Main Menu";
        cout << "\nEnter your choice: ";
        cin >> op;

        switch (op) {
        case 'a':
            cout << a << " > " << b << " = " << (a > b ? "True" : "False") <<
"\n";
            break;
        case 'b':
            cout << a << " < " << b << " = " << (a < b ? "True" : "False") <<
"\n";
            break;
        case 'c':
            cout << a << " >= " << b << " = " << (a >= b ? "True" : "False") <<
"\n";
            break;
```

```
        case 'd':
            cout << a << " <= " << b << " = " << (a <= b ? "True" : "False") <<
"\n";
            break;
        case 'e':
            cout << a << " == " << b << " = " << (a == b ? "True" : "False") <<
"\n";
            break;
        case 'f':
            cout << a << " != " << b << " = " << (a != b ? "True" : "False") <<
"\n";
            break;
        case 'g':
        case 'G':
            relLoop = false;
            break;
        default:
            cout << "Invalid choice!\n";
        }
    }
}

void logicalOperators(int a, int b) {
    char op;
    bool logLoop = true;

    while (logLoop) {
        cout << "\n\n--- Logical Operations ---";
        cout << "\na. && (Logical AND)";
        cout << "\nb. || (Logical OR)";
        cout << "\nc. ! (Logical NOT)";
        cout << "\nd. Exit to Main Menu";
        cout << "\nEnter your choice: ";
        cin >> op;

        switch (op) {
        case 'a':
```

```
            cout << "(" << a << " && " << b << ") = " << (a && b) << "\n";
            break;
        case 'b':
            cout << "(" << a << " || " << b << ") = " << (a || b) << "\n";
            break;
        case 'c':
            cout << "!a = " << (!a) << ", !b = " << (!b) << "\n";
            break;
        case 'd':
        case 'D':
            logLoop = false;
            break;
        default:
            cout << "Invalid choice!\n";
        }
    }
}

void bitwiseOperators(int a, int b) {
    char op;
    bool bitLoop = true;

    while (bitLoop) {
        cout << "\n\n--- Bitwise Operations ---";
        cout << "\na. & (AND)";
        cout << "\nb. | (OR)";
        cout << "\nc. ^ (XOR)";
        cout << "\nd. ~ (NOT)";
        cout << "\ne. << (Left Shift)";
        cout << "\nf. >> (Right Shift)";
        cout << "\ng. Exit to Main Menu";
        cout << "\nEnter your choice: ";
        cin >> op;

        switch (op) {
        case 'a':
            cout << a << " & " << b << " = " << (a & b) << "\n";
```

```
            break;
        case 'b':
            cout << a << " | " << b << " = " << (a | b) << "\n";
            break;
        case 'c':
            cout << a << " ^ " << b << " = " << (a ^ b) << "\n";
            break;
        case 'd':
            cout << "~" << a << " = " << (~a) << ", ~" << b << " = " << (~b) <<
"\n";
            break;
        case 'e':
            cout << a << " << 1 = " << (a << 1) << ", " << b << " << 1 = " << (b
<< 1) << "\n";
            break;
        case 'f':
            cout << a << " >> 1 = " << (a >> 1) << ", " << b << " >> 1 = " << (b
>> 1) << "\n";
            break;
        case 'g':
        case 'G':
            bitLoop = false;
            break;
        default:
            cout << "Invalid choice!\n";
        }
    }
}

int main() {
    int a, b, ch = 0;

    cout << "Enter number 1 and number 2: ";
    cin >> a >> b;

    bool mainLoop = true;
    while (mainLoop) {
```

```
        cout << "\n\n=== Main Menu ===";
        cout << "\n1. Arithmetic Operators";
        cout << "\n2. Relational Operators";
        cout << "\n3. Logical Operators";
        cout << "\n4. Bitwise Operators";
        cout << "\n5. Exit";
        cout << "\nEnter your choice: ";
        cin >> ch;

        switch (ch) {
        case 1:
            arithmeticOperators(a, b);
            break;
        case 2:
            relationalOperators(a, b);
            break;
        case 3:
            logicalOperators(a, b);
            break;
        case 4:
            bitwiseOperators(a, b);
            break;
        case 5:
            cout << "\nExiting program. Goodbye!\n";
            mainLoop = false;
            break;
        default:
            cout << "\nInvalid main menu choice!\n";
        }
    }

    return 0;
}
```

Q.3 no.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()
{
    float per;
    char grade;
    cout << "\nEnter the percentage =";
    cin >> per;
    cout << "\ngrade is =";
    if(per>80)
    {
        cout << "A";
    }
    else if(per>=35 && per<=80)
    {
        cout << "B";
    }
    else
    {
        cout << "C";
    }
    return 0;
}
```

Q.3 no.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"<<endl;
   break;

  }
  else if(guess>47){
```

```
  cout<<"You Guess To high number"<<endl;
 }
 else{
  cout<<"You  Guess To low Number"<<endl;
 }
 count--;

}
 return 0;
}
```

Q.3 no.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 :-

- There are a code of multiplication table of given number :-

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

int main()
{
    int num,i;
    cout << "Enter the number display in multiple table:";
    cin >>num;

    cout << "\n multiplication Table of" << num <<":\n";
```

```
        for(i=0;i<=10;i++)
        {
                cout << num <<" x " << i << " = " << num * i << endl;

        }
        return 0;
    }
```

Q.3 no.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 rows = 5;
   for (int i = 1; i <= rows; i++) {
      for (int j = 1; j <= i; j++) {
         cout << "*";
      }
      cout << endl;
   }
   return 0;
}
```

Q.4 Functions and Scope

No.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 :-

- There are a write a code of simple Calculator :-

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


int add(int a, int b) { return a + b; }
int subtract(int a, int b) { return a - b; }
int multiply(int a, int b) { return a * b; }
int divide(int a, int b) {
   if (b != 0) return a / b;
   else {
      cout << "Error: Division by zero!" << endl;
      return 0;
   }
}

int main() {
   int num1, num2;
   int choice;
```

```
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

    cout << "Choose operation:\n1. Add\n2. Subtract\n3. Multiply\n4.
Divide\n";
    cin >> choice;

    switch(choice) {
        case 1:
                cout << "Result: " << add(num1, num2);
                break;

        case 2:
                 cout << "Result: " << subtract(num1, num2);
                break;
        case 3:
                cout << "Result: " << multiply(num1, num2);
                break;
        case 4:
               cout << "Result: " << divide(num1, num2);
                break;
        default: cout << "Invalid choice!";
    }

    return 0;
}
```

No.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;

}
```

No.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;
int inc_10()
{
        int y;
        num+=200;
        cout<<num;
        return 0;
}
int main()
{
        int x=10;
        cout<<"Before change in number "<<num;
        inc_10();
        cout<<"After change in number "<<num;

}
```

Q.5 Arrays and Strings

No.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 numbers[100];
   float average;
   int n, isum = 0;

   cout << "How many numbers do you want? ";
   cin >> n;

   cout << "Enter " << n << " numbers:\n";
   for(int i = 0; i < n; i++)
   {
      cin >> numbers[i];
      isum += numbers[i];
   }
```

```
    average = (float)isum / n;

    cout << "Sum = " << isum << endl;
    cout << "Average = " << average << endl;

    return 0;
}
```

No.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;
}
```

No.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;
}
```

Q.6 Introduction to Object-Oriented Programming :-

No.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 :-

```
    // simple calculater
#include<iostream>
using namespace std;
```

```
class calculate
{
	public :
		void add(int a,int b)
		{
			cout <<"Adition of two number is :"<<a+b;
		}
		void sub(int a,int b)
		{
			cout <<"Subtraction of two number is :"<<a-b;
		}
		void mul(int a,int b)
		{
			cout <<"multiplication of two number is :"<<a*b;
		}
		void div(int a,int b)
		{
			cout <<"divison of two number is :"<<a/b;
		}
};

int main()
{
	int num1,num2,choice;
	cout <<"Enter the number 1 and 2 :"<<endl;
```

```
cin>>num1>>num2;
calculate c;
cout <<"\n1. Addition :";
cout <<"\n2. subtraction :";
cout <<"\n3. multiplication :";
cout <<"\n4. divison :";

while(choice!=5)
{
	cout <<"\nEnter the choice :";
	cin >> choice;
switch(choice)
{
	case 1:
		c.add(num1,num2);
	break;
	case 2:
		c.sub(num1,num2);
	break;
	case 3:
		c.mul(num1,num2);
	break;
	case 4:
		c.div(num1,num2);
	break;
	case 5:
```

```
                break;
        }
        }

}
```

No.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 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;
 }
};
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;
}
```

No.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= ";
   cin>>name;
  }
  void depart(){
   cout<<"Enter Department Name= ";
   cin>>Department;
  }
  void sub(){
   cout<<"Enter Subject Name= ";
```

```
   cin>>
  }

};
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;
}
```