INDEX

NAME: Saptak Patra STD.: SY-AIDS SEC.: ROLL NO.: 65 SUB.: OOP Practical

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks

HackerRank | Prepare Certify Compete

Search

Saptak Patra

@death_exp69

Complete your profile

Add your missing details →

This data will be helpful to auto-fill your job applications

0%

Personal Information

death.exp69@icloud.com

Add your mobile number

India

My Badges

Problem Solving ★★★★

CPP ★★★★★

Python ★★★★

My Resume

+ Add Resume

Add your resume here

My Certifications

Python (Basic) Verified SKILL

Problem Solving (Basic) Verified SKILL

EEO settings

Privacy - Terms

Experiment-1

1) Write a C++ program to declare a class student having data members as roll no. and name. Accept and display data for single student.

Ans)
```
# <iostream>
using namespace std;
class student
{ int roll_no;
  string name;
  public:
  void accept()
  { cout <<" Enter the student name and roll no: ";
    cin >> name >> roll_no; }
  void disp()
  { cout <<" Student Name:" << name;
    cout <<"\n Student Roll No:" << roll_no; } };
int main()
{ student s1;
  s1.accept();
  s1.disp();
  return 0; }
```

2) Write a C++ code to create a class book having data members as book_name, book_price, and book_id. Accept the data for 2 books and display the name of the book having greater price

Ans)
```
# include <iostream>
using namespace std;
class class_book
```

```
{ string book_name;
  int book_id;
  public:
  int book_price;
  void accept ()
  {
     cout <<"Enter the book name, price and id :";
     cin >> book_name >> book_price >> book_id; }
  void disp ()
  { cout <<"\n Book Name :" << book_name;
    cout <<"\n Book Price :" << book_price;
    cout <<"\n Book ID :" << book_id; } };
  int main ()
  { class_book B1, B2;
    B1.accept();
    B2.accept();
    if (B1.book_price > B2.book_price)
    { B1.disp();
    else
    { B2.disp (); }
     return 0; }
```

3) Write a C++ program to declare a class time having data members as H, M and S. Accept data for one object and display total time in seconds

Ans)
```
# include <iostream>
using namespace std;
class Time
{ int H, M, S;
  public:
  void accept ()
```

```
{ cout << " Enter the time in hours minutes seconds: ";
  cin >> H >> M >> S; }
void calculate()
{ H = H * 3600;
  M = M * 60;
  S = S + H + M; }
void disp()
{ cout << " Total time in seconds = " << S; } };
int main()
{ Time T1;
  T1.accept(); T1.calculate();
  T1.disp()
  return 0; }
```

5/8

Experiment - 2

1) WAP to declare a class 'city' having data members as name and population. Accept this data for 5 cities and display name of city having highest population.

Ans)
```
# include <iostream>
using namespace std;
class city
{ public:
  string name;
  int population;
  void accept()
  { cout <<"Enter City Name : ";
   cin >> name;
   cout <<"Enter City Population:";
  cin >> population; }
  void disp()
  { count <<"City having highest population:" << name;}
  int main()
  { city C[5];
    int i, max;
    for (i=0; i<5; i++)
    { C[i].accept(); }
    max = C[0].population;
   for (i=0; i<5; i++)
   { if (C[i].population > max)
    { max = i } }
    C[max].disp();
   return 0; }
```

2) Wap to declare a class 'Account' having data members as Account and balance. Accept this data for 10 accounts and give interest of 10% where balance is equal or greater than 5000 and display them.

Ans)
```
# include <iostream>
class account
{ public:
  int acc_no;
  float balance;
  void accept()
  { count << " Enter Account Number: ";
    cin >> acc_no;
    count << " Enter Account Balance: ";
    cin >> balance; }
  void disp()
  { count << "\n Account Number :" << acc_no;
    count << "\n Account Balance :" << balance; }};
int main ()
{ account A[10];
  int i;
  for (i=0; i<10; i++)
  { A[i]. accept(); }
  for (i=0; i<10; i++)
  { if (A[i]. balance >= 5000)
  { A[i]. balance = A[i]. balance + (0.1 * A[i]);
    A[i]. balance);
    A[i]. disp(); }}
  return 0; }
```

3) Wap to declare a class 'staff' having data members as name and post. Accept this data for 5 staff and display names of staff who are "HOD".

Ans:
```
#include <iostream>
using namespace std;
class staff
{ string name;
  public:
  string post;
  void accept()
  { cout << "Enter the staff name :";
    getline (cin, name);
    cout << "Enter the staff post :";
    getline (cin, post); }
  void disp()
  { cout << "\n Staff Name :" << name;
    cout << "\n Staff Post :" << post; }};
int main()
{ staff s[5];
  int i;
  for (i=0; i<5; i++)
  { s[i].accept(); }
  for (i=0; i<5; i++)
  { if (s[i].post == "HOD")
  { s[i].disp(); }}
  return 0; }
```

5/8/25

Experiment-3

1) Write a program to declare a class 'book' containing data members as book_title, author_name and price. Accept and display the information for one object using a pointer to that object.

Ans)
```
#include <iostream>
using namespace std;
class book
{ string booktitle;
  string authorname;
  int price;
  public:
  void accept()
  { cout << "Enter Book Title: ";
    getline (cin, booktitle);
    cout << " Enter Author name: ";
    getline (cin, authorname);
    cout << " Enter Book Price: ";
    cin >> price; }
  void disp()
  { cout << "Book Title: " << booktitle;
    cout << "\n Author Name: " << authorname;
    cout << "\n Book Price: " << price; } };
int main()
{ book B1;
  book * p;
  p = & B1;
  p -> accept();
  p -> disp();
  return 0; }
```

2) WAP to declare a class 'student' having

Ans)
```
#include <iostream>
using namespace std;
class student
{ int roll_no;
  float percentage;
  public:
  void accept()
  { cout << "Enter the Student Roll No:";
    cin >> roll_no;
    cout << "Enter Student Percentage:";
    cin >> percentage; }
    void disp()
    { this -> accept();
      cout << "\n Roll No of the student :" << roll_no;
      cout << "\n Percentage of the student :" << percentage;
      }};
int main()
{ student S1;
  S1 disp();
  return 0;
```

3) Write a program to demonstrate the use of nested class?

Ans)
```
#include <iostream>
using namespace std;
class student
{
    private:
```

```
        String name;
        int roll_no;
        public:
        void inputStudentDetails ()
        {
            cout <<"Enter Student's name :";
            getline (cin, name);
            cout <<"Enter roll number :";
            cin >> roll no;
        }
        void displayStudentDetails ()
        {
            cout <<"\n Student Name :"<< name <<endl
            cout <<"Roll No : "<< roll_no << endl;
        }
class Marks
{
    private:
    int science;
    int english;
    public:
    void inputMarks ()
    {
        cout <<"Enter marks in science :";
        cin >> science;
        cout <<"Enter marks in English :";
        cin >> english;
    }
    void displayMarks ()
    {
        cout <<"Marks in Science :"<< science << endl;
        cout <<"Marks in English :"<< english << endl;
```

```
        }
    };
};
int main ()
{
    Student S1;
    Student :: Marks M1;
    S1.inputStudentDetails ();
    M1.inputMarks ();
    S1.displayStudentDetails ();
    M1.displayMarks ();
    return 0;
}
```

28/5

Practice Questions

1) Create two classes, Class A and Class B, each with a private integer. Write a friend function sum () that can access private data from both classes and return the sum.

Ans)
```
# include <iostream>
using namespace std;
class Class B;
class Class A {
    int A;
public:
void accept () {
    count <<" Enter value for A: ";
    cin >> A;
}
friend int sum (Class A oA, Class oB);
} na;
class Class B {
    int B;
public:
void accept () {
    count <<" Enter value for B: ";
    cin >> B;
}
friend int sum ( Class A oA, Class B oB);
} nb;
int sum ( Class A oA, Class oB) {
    return oA.A + oB.B;
}
int main () {
    na. accept ();
```

```
    nb.accept();
    cout << " Sum : " << sum(na, nb) << endl;
    return 0;
}
```

2) Write a program with a class Number that contains a private integer. Use a friend function swapNumbers (Number &, Number &) to swap the private values of two Number objects.

Ans)
```
#include <iostream>
using namespace std;
class Number {
    int num;
public:
    void accept() {
        cout << "Enter value : ";
        cin >> num;
    }
    void display() {
        count << "Value : " << num << endl;
    }
    friend void swapNumbers (Number & num1,
                             Number & num2);
} n1, n2;
void swapNumbers (Number &, Number &) {
    int temp = n1.num;
    n1.num = n2.num;
    n2.num = temp;
}
int main() {
    n1.accept();
```

```
    n2.accept();
    cout << "Before swap:" << endl;
    n1.display();
    n2.display();
    swapNumbers(n1, n2);
    cout << "After swap:" << endl;
    n1.display();
    n2.display();
    return 0;
}
```

3) Define two classes Box and Cube, each having a private volume. Write a friend function findGreater(Box, Cube) that determines which object has a larger volume.

Ans)
```
#include <iostream>
using namespace std;
class Cube;
class Box {
    double v;
public:
    void accept() {
        double l, w, h;
        cout << "Enter length, width, and height for
                the Box: ";
        cin >> l >> w >> h;
        v = l * w * h;
    } friend void findGreater(Box b, Cube c);
} b;
class Cube {
    double v;
public:
```

```
    void accept() {
        double s;
        cout << "Enter side length for the Cube:";
        cin >> s;
        v = s * s * s;
    } friend void findGreater(Box b, Cube c);
} c;
void findGreater(Box, Cube) {
    string res = (b.v > c.v) ? "Box" : ((c.v > b.v) ?
    "Cube" : "Both equal.");
    cout << "Greater Volume: " << res << endl;
}
int main() {
    b.accept();
    c.accept();
    findGreater(b, c);
    return 0;
}
```

4) Create a class Complex with real and imaginary parts as private members. Use a friend function to add two complex numbers and return the result as a new Complex object.

Ans)

```
#include <iostream>
using namespace std;
class Complex {
    double r, i;
public:
    void accept() {
        cout << "Enter real and imaginary parts: ";
        cin >> r >> i;
    }
```

```
    void display () {
        cout << r << " + " << i << " ;" << endl;
    } friend Complex add ( Complex c1, Complex c2);
} c1, c2;
Complex add (Complex, Complex) {
    Complex sum;
    sum.r = c1.r + c2.r;
    sum.i = c1.i + c2.i;
    return sum;
}
int main () {
    c1.accept ();
    c2.accept ();
    Complex sum = add (c1, c2);
    cout << " The sum is: ";
    sum.display ();
    return 0;
}
```

5) Create a class Student with private data members: name and three subject marks. Write a friend function calculate-Average (Student) that calculates and display the average marks.

Ans)
```
#include <iostream>
using namespace std;
class Student {
    string n;
    int m[3];
public:
    void accept () {
        cout << "Enter student name: ";
        cin >> n;
```

```
        cout <<"Enter marks for three subjects: ";
        cin >> m[0] >> m[1] >> m[2];
    } friend void calculateAverage (Student s);
} s;
void calculateAverage (Student) {
    double avg = (s.m[0] + s.m[1] + s.m[2]) / 3.0;
    cout <<"Student: " << s.n << endl;
    cout <<"Average Marks: " << avg << endl;
}
int main () {
    s.accept();
    calculateAverage (s);
    return 0;
}
```

6) Create three classes: Alpha, Beta, and Gramma, each with a private data member. Write a single friend function that can access all three and print their sum.

Ans)
```
# include <iostream>
using namespace std;
class Beta;
class Gramma;
class Alpha {
int n; public:
    void accept () {
        cout << "Enter value for Alpha: ";
        cin >> n;
    } friend int sum (Alpha a, Beta b, Gramma c);
};
class Beta {
int n; public:
```

```
    void accept() {
        cout << "Enter value for Beta: ";
        cin >> x;
    } friend int sum (Alpha a, Beta b, Gamma c);
} b;
class Gamma {
int x; public:
    void accept() {
        cout << "Enter value for Gamma:";
        cin >> x;
    } friend int sum (Alpha a, Beta b, Gamma c);
} c;
int sum (Alpha a, Beta b, Gamma c) {
    return a.x + b.x + c.x; }
int main() {
    a.accept();
    b.accept();
    c.accept();
    cout << "Sum: " << sum(a, b, c) << endl;
    return 0;
}
```

7) Create a class Point with private members x and y. Write a friend function that calculates and returns the distance between two Point objects.

Ans)
```
# include <iostream>
# include <cmath>
using namespace std;
class Point {
    double x, y;
public:
```

```
    void accept () {
        cout << "Enter x and y coordinates: ";
        cin >> x >> y;
    } friend double dist (Point p1, Point p2);
} p1, p2;
double dist (Point, Point) {
    double dx = p1.x - p2.x;
    double dy = p1.y - p2.y;
    return sqrt ((dx * dx) + (dy * dy));
}
int main () {
    p1.accept ();
    p2.accept ();
    cout << "Distance: " << dist (p1, p2) << endl;
    return 0;
}
```

8) Create two classes: BankAccount and Audit. BankAccout holds private balance information. Write a friend function in Audit that accesses and prints balance information for auditing.

Ans)
```
# include <iostream>
using namespace std;
class BankAccount;
class Audit {
public:
    void pBal (BankAccount);
} and;
class BankAccount {
double b; public:
    void accept () {
```

```
    cout << "Enter bank account balance : ";
    cin >> b;
  } friend void Audit::pBal(BankAccount);
} ba;
void Audit::pBal(BankAccount) {
    cout << "Auditing .... Bank Account balance : ₹"
         << ba.b << endl;
}
int main() {
    ba.accept();
    aud.pBal(ba);
    return 0;
}
```

Experiment 4

Q) Write a C++ program to resolve the following problem statements and show output.

1) Swap two numbers from same class using object as function argument. Write swap function as member function.

=>
```
# include <iostream>
using namespace std;
class num {
public:
    int n1, n2;
    void accept () {
        cout <<"Enter first number:";
        cin >>n1;
        cout <<" Enter second number: ";
        cin >>n2;
    }
    void display () {
        cout <<" Num 1: " << n1 << endl;
        cout <<" Num 2: " << n2 << endl;
    }
    void swap (num &obj) {
        int temp = obj.n1;
        obj.n1 = obj.n2;
        obj.n2 = temp;
        cout << "Swapped :-\n";
        cout << "First number: " << obj.n1 << endl;
        cout << " Second number : " << obj.n2 << endl;
    }
};
```

```
int main () {
    num nu
    nu. accept ();
    nu. display ();
    nu. swap (nu);
    return 0;
}
```

· Output :-

```
Enter first number : 5
Enter second number : 9
Num 1 : 5
Num 2 : 9
Swapped :-
First number : 9
Second number : 5
```

2) Swap two numbers from same class using concept of friend function.

=>
```
# include <iostream>
using namespace std;
class num {
public:
    int n 1, n 2;
    void accept () {
        cout << " Enter first number : ";
        cin >> n 1;
        cout << " Enter second number : ";
        cin >> n 2;
    }
```

```
    void display () {
        cout << "Num 1: " << n1 << endl;
        cout << "Num 2: " << n2 << endl;
    }
    friend void swap (num & nu);
};
void swap (num & nu) {
    int temp = nu.n1;
    nu.n1 = nu.n2;
    nu.n2 = temp;
    cout << "Swapped: \n";
    cout << "First: " << nu.n1 << endl;
    cout << "Second: " << nu.n2 << endl;
}
int main () {
    num nu;
    nu.accept();
    nu.display();
    swap (nu);
    return 0;
}
```

- Output :-

```
Enter first number : 5
Enter second number : 9
Num 1: 5
Num 2: 9
Swapped:
First : 9
Second : 5
```

3) Swap two numbers from different class using friend function.

=>
```
# include <iostream>
using namespace std;
class num 1;
class num 2;
class num 1 {
public:
    int n 1;
    void accept () {
        cout << "Enter first number : ";
        cin >> n 1;
    }
    void display () {
        cout << " Num 1 : " << n 1 << endl;
    }
    friend void swap (num 1 & nu 1, num 2 & nu 2);
};
class num 2 {
public:
    int n 2;
    void accept () {
        cout << "Enter second number : ";
        cin >> n 2;
    }
    void display () {
        cout << " Num 2 : " << n 2 << endl;
    }
    friend void swap (num 1 & nu 1, num 2 & nu 2);
};
void swap (num 1 & nu 1, num 2 & nu 2) {
    int temp = nu 1. n 1;
```

```
        nu1.n1 = nu2.n2;
        nu2.n2 = temp;
        cout << "Swapped: \n";
        cout << "First: " << nu1.n1 << endl;
        cout << "Second: " << nu2.n2 << endl;
}
int main () {
    num1 nu1;
    num2 nu2;
    nu1.accept();
    nu2.accept();
    nu1.display();
    nu2.display();
    swap(nu1, nu2);
    return 0;
}
```

- Output:-

```
Enter first number: 5
Enter second number: 9
Num 1: 5
Num 2: 9
Swapped:
First: 9
Second: 5
```

4) Create two classes Result 1 and Result 2 which stores the marks of the students. Read the value of marks for both the class objects & compute average.

=>
```
# include <iostream>
using namespace std;
class Result 1 {
public:
    int r 1;
    void accept () {
        cout << " Enter first result :";
        cin >> r 1;
    }
} res 1;
class Result 2 {
public:
    int r 2;
    void accept () {
        cout << " Enter second result :";
        cin >> r 2;
    }
} res 2;
float avg (const Result 1 & res 1, const Result 2 & res 2) {
    return (res 1.r 1 + res 2.r 2) / 2.0f;
}
int main () {
    res 1.accept ();
    res 2.accept ();
    float av = avg (res 1, res 2);
    cout << " Average : " << av;
    return 0;
}
```

• Output:-

Enter first result : 95
Enter second result : 99
Average : 97

5) Find the greatest number among two numbers from two different classes using friend.

=>
```
#include <iostream>
using namespace std;
class num1;
class num2;

class num1 {
public:
    int n1;
    void accept() {
        cout <<" Enter first number : ";
        cin >>n1;
    }
    friend int gnum (const num1 &, const num2 &);
} n1;
class num2 {
public:
    int n2;
    void accept() {
        cout <<" Enter second number : ";
        cin >>n2;
    }
    friend int gnum (const num1 &, const num2 &);
} n2;
```

```
int grum (const num1 & a, const num2 & b) {
    return (a.n1 > b.n2) ? a.n1 : b.n2;
}
int main () {
    nu1.accept ();
    nu2.accept ();
    int gr = grum (nu1, nu2);
    cout << "Greatest : " << gr;
    return 0;
}
```

Output :-

```
Enter first number : 5
Enter second number : 9
Greatest : 9
```

28/8

Experiment - 5

a) WAP to find the sum of numbers between 1 to n using a constructor where the value of n will be passed to the constructor.

Ans)
```
# include <iostream>
using namespace std;
class Sum
{
    int n, sum;
    public:
        Sum (int num)
        {
            n = num;
            sum = 0;
            for (int i = 0; i < n; i++)
            {
                sum += i;
            }
            cout << " Sum = " << sum;
        }
};
int main ()
{
    Sum s(10);
    return 0;
}
```

b) WAP to declare a class 'student' having data members as name and percentage. Write a constructor to initialize the data members. Accept & display data for one student.

Ans)
```
# include < iostream >
# include < string >
using namespace std;
class student
{
    private:
    String name;
    float percentage;
    public:
        student ( String n, float p)
        {
            name = n;
            percentage = p;
            cout << " Name : ";
            getline ( cin, name);
            cout << " Percentage : ";
            cin >> percentage;
        }
        void display ()
        {
            cout << " Student Name : " << name << endl;
            cout << " Student Percentage : " << percentage
                                                << endl;
        }
};
int main () {
    Student S1 ("Saptak", 87.5)
    S1. display ();
    return 0;
}
```

c) Define a class 'college', members variables as roll_no, name, course, WAP using constructor with default value as "Computer Engineering" for course. Accept this data for two objects of class and display the data.

Ans)
```
#include <String>
#include <iostream>
using namespace std;
class College
{
    private:
    String course_name, stud_name;
    int roll_no;
    public:
    College ()
    {
        roll_no = 65;
        stud_name = "Saptak";
        course_name = "C.S.E";
    }
    void display ()
    {
        cout << "Roll no : " << roll_no << endl;
        cout << "Student name : " << stud_name << endl;
        cout << "Course name : " << course_name << endl;
    }
};
int main ()
{
    College C1;
    C1.display ();
    return 0;
}
```

d) Write a program to demonstrate constructor overloading.

Ans)
```
# include <iostream>
using namespace std;
class rectangle
{
    int l, b;
    public:
    rectangle()
    {
        l = 2;
        b = 5;
    }
    rectangle (int x)
    {
        l = x;
        b = x;
    }
    rectangle (int x, int y)
    {
        l = x;
        b = y;
    }
    rectangle (rectangle & r3)
    {
        l = r3.l;
        b = r3.b;
    }
    void calculate ()
    {
        cout << "The area is : " << (l * b) << endl;
    }
};
```

```
int main()
{
    rectangle r1;
    r1.calculate();
    rectangle r2(3);
    r2.calculate();
    rectangle r3(6,6);
    r3.calculate();
    rectangle r4(r3);
    r4.calculate();
    return 0;
}
```

10/11

Experiment-6

a) WAP to implement multi level inheritance. Assume suitable data.

Ans)
```
# include <iostream>
# include <string>
using namespace std;
class Person
{
    protected:
    int age;
    String name;
};
class Student:public Person
{
    private:
    int roll_no;
    public:
    void accept()
    {
        cout << "Enter your name: ";
        cin >> name;
        cout << "Enter your age: ";
        cin >> age;
        cout << "Enter your roll no: ";
        cin >> roll_no;
    }
    void display()
    {
        cout << "Name:\n" << name;
        cout << "Age:\n" << age;
        cout << "Roll No:\n" << roll_no;
    }
```

```
};
int main ()
{
    Student S1;
    S1.accept ();
    S1.display ();
    return 0;
}
```

b) Write a program to implement multiple classes. Assume suitable data.

Ans)

```
#include <iostream>
using namespace std;
class Academic
{
    protected :
    int marks;
};
class Sports
{
    protected :
    int score;
};
class Result : protected Academic, protected Sports
{
    int total_score = 0;
    public :
    void accept ()
    {
        cout <<" Enter the marks of the Student: ";
        cin >> marks;
```

```
    cout << " Enter the sports score of the student : ";
    cin >> score;
  }
  void calculate ()
  {
      total score = marks + score;
      cout << " The marks of the student is : " << marks
          << endl;
      cout << " The sports score of the student is : <<
          score << endl;
      cout << " The total score of the student is : <<
          total_score << endl;
    }
};
int main ()
{
    Result r;
    r.accept ();
    r.calculate ();
    return 0;
}
```

c) WAP to implement hierarchical inheritance. Assume suitable data.

Ans)
```
#include <iostream>
using namespace std;
class Vehical
{
    prote
    String brand;
    int model;
};
class car : protected vehical
{
    protected :
        String type;
};
class Electric Car : protected Car
{
    int battery capacity;
    public :
    void accept ()
    {
        cout << " Enter the brand of car : ";
        cin >> brand;
        cout << " Enter the model of the car : ";
        cin >> model;
        cout << " Enter the type of car : ";
        cin >> type;
        cout << " Enter the battery capacity of the car : ";
        cin >> battery capacity;
    }
```

```
void display ()
{
    cout << "\n The brand of the car : " << brand
                                        << endl;
    cout << " The model of the car : " << model << endl;
    cout << " The type of the car : " << type << endl;
    cout << " The battery capacity of the car : <<
            battery capacity << endl;
}
};
int main ()
{
    Electric car e;
    e. accept ();
    e. display ();
    return 0;
}
```

d) WAP to implement hybrid inheritance

Ans)
```
#include <string>
#include <iostream>
using namespace std;
class Person
{
    public:
    String name;
    int age;
    void get Personal Details ()
    {
        cout <<" Enter name :";
        cin >> name;
        cout <<" Enter age :";
        cin >> age;
    }
    void show Details ()
    {
        cout <<" Name : "<<name <<", Age :"<<age
            << endl;
    }
};
class Student public Person
{
    public:
    String course;
    void get Details ()
    {
        cout <<" Enter course : ";
        cin >> course;
    }
```

```
        void show details()
        {
            cout <<" Course: " << course << endl;
        }
};
class Employee : public Person
{
    public:
    String company;
    void getemployee details ()
    {
        cout <<" Enter company: ";
        cin >> company;
    }
    void Show Employee details ()
    {
        cout <<"Company: " << company << endl;
    }
};
class Intern : public Student , public Employee
{
    public:
    void Show Intern details ()
    {
        cout <<" \n ---- Intern Details ---- \n ";
        Student :: Show Person details ();
        Show Employee details ();
    }
};
int main ()
{
    Intern it;
```

```
it.Student::getPersonalDetails();
it.getDetails();
it.getEmployeeDetails();
it.showInternDetails();
return 0;
}
```

10/11

Experiment 7

a) WAP using function overloading to calculate the area of a laboratory (rectangular in shape) & area of classroom (square).

Ans)
```
#include <iostream>
using namespace std;
class Area;
{
    private:
    float l, b;
    public:
    void area (float l)
    {
        float area;
        area = l * l;
        cout <<" Area of classroom : " << area << endl;
    }
    void area (float l, float b)
    {
        float area;
        area = l * b;
        cout << " Area of laboratory : " << area;
    }
};
int main ()
{
    Area a 1;
    a 1. area (8);
    a 1. area (4,12);
    return 0;
}
```

b) WAP using function overloading to calculate the sum of 5 float values & 10 integer values.

Ans)
```
# include <iostream>
using namespace std;
class Sum
{
    private:
    int a, b, c, d, e, f, g, h, i, j;
    float k, l, m, n, o;
    public:
    void sum (float k, float l, float m, float n, float o)
    {
        float sum;
        sum = k + l + m + n + o;
        cout <<" Sum of 5 floating numbers : " << sum
        << endl;
    }
    void sum ( int a, int b, int c, int d, int e, int f,
        int g, int h, int i, int j)
    {
        int sum;
        sum = a + b + c + d + e + f + g + h + i + j;
        cout <<" Sum of 10 integers :" << sum;
    }
};
int main ()
{
    Sum s1;
    s1. sum (1.2, 3.5, 5.6, 8.4, 1.5)
    s1. sum ( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    return 0;
}
```

c) WAP to implement unary operator when used with the object so that the numeric data member of the class is negated.

Ans)
```
#include <iostream>
using namespace std;
class Number
{
    private: int x;
    public void accept()
    {
        cout << "Enter a number : ";
        cin >> x;
    void operator - ()
    {
        x = -x;
    }
    void display ()
    {
        cout << "Negated number : " << x << endl;
};
int main ()
{
    Number n1;
    n1.accept();
    -n1;
    n1.display();
    return 0;
}
```

d) WAP to implement the unary ++ operator (for pre increment and post increment) when used with the object so that the numeric data member of the class is incremented

Ans)
```
#include <iostream>
using namespace std;
class Number
{
    private: int x; int accept; int temp;
    public:
    void accept()
    {
        cout << " Enter a number ";
        cin >>x; temp=x;
    }
    void operator ++()
    {
        x=++x;
    }
    void reset()
    {
        x= temp;
    }
    void operator ++(int)
    {
        x=x++
    }
    void display ()
    {
        cout << " (pre) The number is : "<< x << endl;
    }
```

```
void display2()
{
    cout << "(post) The number is : " << x;
}
};
int main()
{
    Number n1;
    n1.accept();
    ++n1;
    n1.display();
    n1.reset();
    n1++;
    n1.display2();
    return 0;
}
```

10/11

Experiment - 8

a) WAP to overload '+' operator so that two strings can be concatenated. Eg: "xyz" + "pqr" = "xyz pqr".

Ans)
```
# include <iostream>
# include <string>
using namespace std;
class mystring
{       char str[20];
        public:
        mystring (char a[20])
        {
                strcpy (str, a);
        }
        void display()
        {
                cout<<" String is " << str <<"\n";
        }
        my string operator +(mystring);
};
mystring mystring :: operator +(mystring s2)
{       mystring s3;
        strcpy (s3.str, str);
        strcat (s3.str, s2.str);
        return s3;
}
int main ()
{       mystring s1 ("abc"); mystring s2("xyz");
        s3 = s1 + s2;
        cout <<"\n concatenated ";
        s3.display();
        return 0;
}
```

b) Wap to create base class I login --:-

```
# include <iostream>
using namespace std;
class ilogin
{
    protected:
        char n[20], p[20];
    public:
        virtual void accept ()
        {   cout << " Enter name : ";
            cin >> n;
            cout << " Enter password : ";
            cin >> p;
        }
        virtual void public ()
        {   cout << " \n name is : " << n;
            cout << " \n password is : " << p;
        }
};
class email_login : public ilogin
{
    public:
        void accept ()
        {   cout << " Enter email id : ";
            cin >> n;
            cout << " Enter email-id password : ";
            cin >> p;
        }
        void display ()
        {   cout << " \n email id is : " << n;
            cout << " \n email id password is : " << p;
        }
};
```

```
class membership_login : public ilogin
{   public:
        void accept()
        {   cout<<"Enter membership name : ";
            cin >>n;
            cout <<"Enter membership password : ";
            cin >>p;
        }
        void display ()
        {   cout<<"\n membership name is : "<<n;
            cout <<"\n membership password is : "<<p;
        }
};
int main ()
{       ilogin *ill, ill;
        email_login el;
        membership_login ml;
        int type;
        cout <<"\n 1:- for email login \n 2:- for
                membership login \n 3:- normal login : ";
        cin >> type;
        if (type == 1)
        {   i1 = & el;
            i1 -> accept ();
            i1 -> display ();
        }
        else if (type == 2)
        {   i1 = & ml;
            i1 -> accept ();
            i1 -> display ();
        }
        else if (type == 3)
```

```
{ i1 = &i11;
  i1 -> accept();
  i1 -> display();
}
return 0;
}
```

10/11

Experiment 9

1) WAP to copy the contents of one file to another

```
# include < iostream >
# include < fstream >
using namespace std ;
int main ()
{
    if stream f1;
    ofstream f2;
    char ch;
    f1.open ("myfile.txt", ios::in);
    f2.open ("abc.txt", ios::out);
    while (( ch == fi.get()) != EOF)
    {
        f2.put( ch);
    }
    f1.close ();
    f2.close ();
    cout << " contents copied : ";
    return 0;
}
```

b) WAP to count digits and spaces using file handling

```
# include <iostream>
# include <fstream> # include <ctype.h>
using namespace std;
int main()
{
    ifstream f1("myfile.txt");
    char ch;
    int d=0 ; s=0;
    while ((ch=f1.get())!=EOF)
    {
        if (isdigit(ch))
        {
            d++;
        }
        else if(isspace(ch))
        {
            s++;
        }
    }
    cout <<"\n number of digits in file are: "<< d;
    cout <<"\n number of spaces in file are: "<< s;
    return 0;
}
```

c) WAP to count words using File handling.

```
# include < iostream >
# include < fstream >
# include < ctype.h >
using namespace std;
int main ()
{
        ifstream f1("myfile.txt");
        char ch;
        int w=0;
        while (!ch=f1.get())!=EOF)
        {
            if(isspace(ch))
            {
                w++;
            }
        }
        cout <<" Total words are : " << w;
        return 0;
}
```

d) WAP to count occurences of a given word using file handling

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ifstream f1("myfile.txt");
    string ch;
    int w=0;
    cout <<" Enter a word :");
    cin >> ch;
    string word;
    while (f1 >> word)
    {
        cout <<"\n" << word;
        if ( ch == word)
        {
            w++;
        }
    }
    f1.close();
    cout <<"\n Occurrence of word in file is :" << w;
    return 0;
}
```

10/11

Experiment 10

i) WAP to find sum of array using function template

```
# include <iostream>
using namespace std;
template <class t>
t sum ( t arr [10])
{
    int i;
    t add =0;
    for (i=0; i<10; i++)
    {
        add = add + arr [i];
    }
    return add;
}
int main ()
{
    float arr [ ] = { 12, 1, 3, 5, 6, 3, 4, 9, 10, 14, 15 };
    cout << "\n sum of array elements is : " << sum
                                                ( arr);
    return 0;
}
```

ii) Write a C++ program of square function using template specialization

```
# include <iostream>
using namespace std;
template < class t > t square ( t num)
{
    return num * num;
}
template < > string square ( string str)
{
    return str * str;
}
int main ()
{
    int n = 5;
    string t = " hello ";
    cout << " square of no. is : " << square ( n);
    cout << " square of string is : " << square ( t);
    return 0;
}
```

iii) WAP to build simple calculator using class template.

```
#include <iostream>
Using namespace std;
template <class T> class c
{
    public: T num1=0; T num2=0;
    void accept ()
    {
        cout << "Enter nos :";
        cin >> num1 >> num2;
    }
    void add() {
        cout << "Sum =" << num1 + num2 << endl;
    }
    void sub() {
        cout << "Difference =" << num1 - num2 << endl;
    }
    void mult () {
        cout << "Product =" << num1 * num2 << endl;
    }
    void div() {
        if (num2 != 0)
            cout << "Division =" << num1 / num2 << endl;
        else
            cout << "Error: Division by zero not
                    allowed." << endl;
    }
};
int main () {
    c.<int> obj;
```

```
obj.add();
obj.sub();
obj.mult();
obj.div();
return 0;
}
```

10/11

Experiment 11

1) Without itrator:-

```
→ # include <iostream>
# include <vector>
# include <cctype>
Using namespace std;
int main() {
    vector <char> v(10);
    unsigned int i;
    cout <<" size = " << v.size() << " " << endl;
    for (i=0; i <10; i++) { v[i] = i + 'a'; }
    cout << " Current content: \n";
    for (i=0; i< v.size(); i++) { cout <<v[i] << " "; }
    cout << "\n\n"; cout << " expanding vector: \n";
    for (i=0; i<10; i++) { v.push_back (i+10+'a'); }
    cout <<" size now = " << v.size() << endl;
    cout << " Current contest: \n";
    for(i=0; i<v.size(); i++) { cout << v[i] << " "; }
    cout << "\n\n";
    for (i=0; i <v.size(); i++) { v[i] = toupper (v[i]); }
    cout << " Modified contents: \n";
    for (i=0; i < v.size(); i++) { cout << v[i] << " "; }
    cout << endl;
    return 0;
}
```

Q2) with iterator :-

→
```
# include <iostream>
# include <vector>
Using namespace std;
int main () {
    vector <char> v(10);
    vector <char> :: iterator p;
    int i;
    p = v. begin ();
    i = 0;
    while (p != v.end()) {
        * p = i + 'a';
        p ++;
        i ++;
    }
    cout << "Original content : \n";
    p = v. begin ();
    while (p != v. end ()) {
        cout << * p << " ";
        p++;
    }
```

10/11

Experiment 12

Q1) Implement Stack

→
```
# include <bits/std.c++.h>
using namespace std;
int main(){
    Stack <char> cars;
    cars.push("BMW"); cars.push("Audi");
    cars.push("Mercedes"); cars.push("Ferrari");
    cout << "Top element is :" << cars.top() << endl;
    cout << "size of stack is :" << cars.size() << endl;
    cars.pop(); cars.pop();
    while (! cars.empty()){
        cout << "Element in stack are :" << cars.top()
                                << endl;

        car.pop();
    }
    return 0;
}
```

· Output :-

Top element : Ferrari
Size of stack is : 4
Element in stack are : Audi
Elements in stack are : BMW

2) Implement Queue :-

→
```
# include <bits/stdc++.h>
using namespace std;
int main(){
    queue <int> age;
    age.push(21); age.push(22); age.push(23);
    age.push(24);
    cout << "front element is : " << age.front()<<endl;
    cout << " Back element is : " << age.back()<<endl;
    age.pop(); age.pop();
    while (!age.empty())
    {
        cout << " Element in queue are : " << age
                                front()<<endl;
        age.pop();}
    return 0;
}
```

- O/P :-

front element is : 21
back element is : 24
Elements in queue are : ~~23~~ 23
Elements in queue are : 24

10/4