OOPs with C++ Laboratory

3rd SEM ISE.

INDEX

1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

2. Write a C++ Program to accept an alphabet and check whether it is a vowel or a consonant. If it is a vowel, return its predecessor, else its successor. Use call-by-reference with reference arguments

3. Write a C++ Program to call a C function using an extern "C" linkage directive.Use compound statement linkage directive for #include<math.h>.

4. Write a C++ Program to accept a line of text and count the number of words,characters and digits in it.

5. Write a C++ Program to store two binary numbers in arrays and perform bitwise AND, OR and XOR operations on these two numbers.

6. Create two classes D1 and D2 which store the value of distances. D1 stores distances in meters and centimeters and D2 in feet and inches. Write a program to add objects of two classes D1 and D2 and the display the results in feet and inches using friend function.

7. Given that an EMPLOYEE class contains the data members like E_Number, E_Name, Basic_salary, DA, HRA, Net_salary and the member functions like Read(), Calculate_Net_Sal(), and Display(). Write a C++ Program to read the data of N Employees and Compute the Net_Salary of each employee.

8. Write a C++ program to overload the function Search() to search an integer key value and a key value of type double.

9. Write a C++ program to find the following using Function Template
a) Successor value of any input of type integer, float, char and double.

b) Sum of all the elements of an array of integers or floats or doubles.

10.Write a C++ Program to create a class as COMPLEX and implement the following by overloading the function ADD() which returns the Complex numbers

 a) ADD(C1, C2); C1 is an integer ; C2 is a Complex number.

 b) ADD(C1, C2); C1 and C2 are Complex numbers.

11. Write a C++ Program to create a class called DATE. Accept two valid dates in the form of DD/MM/YYYY. Implement the following by overloading the operators – (minus) and + (Plus). Display the results by overloading the operator << after every operation.

a) No_of _days = D1–D2

b) D1=D1 + No_of _days

 where d1 and d2 are date objects, d1>d2 and No_of_days is an integer.

12. Write a C++ program to create a class called MATRIX using a two-dimensional array of integers. Implement the following by overloading the operator == which checks the compatibility of two matrices to be added and subtracted. Perform the following by overloading + and – operators. Display the result by overloading the operator <<.

13. Write a C++ program to create a class called STUDENT with data members USN,Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester,Fees and Stipend. Enter the data for at least 5 students. Find the semester wise average age for all UG and PG students separately.

14. Write a C++ program to create a base class Student, from this inherit a new class called Exams, containing Marks1, Marks2 and Marks3 as its data members. Also create another class called Sports having Sports_grade as its data member. Now, create yet another class but a derived class of Exams and Sports classes and call it as Awards. Use appropriate member functions in all classes. You can use the Regno, Name, Semester and Branch as the members of the base class.
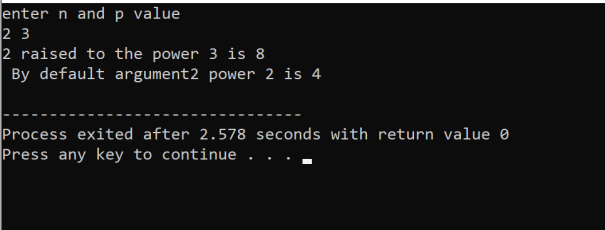
15. Write a C++ program to create a file called File1.txt and print its contents in the reverse order to the File2.txt.

1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

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

double power(double n, int p=2)
{
    double res=1;
    for(int i=1; i<=p;i++)
    res=res*n;
    return res;

}

int main()
{
    double n,result;
    int p;
    cout<<"enter n and p value"<<endl;
    cin>>n>>p;
    result=power(n,p);
    cout<<n<<" raised to the power "<<p<<" is "<<result<<endl;
    result=power(n);
    cout<<" By default argument"<<n<<" power 2 is "<<result<<endl;
    return 0;
}
```

```
 C:\Users\user\Desktop\2.exe
enter n and p value
2 3
2 raised to the power 3 is 8
 By default argument2 power 2 is 4

------------------------------
Process exited after 2.578 seconds with return value 0
Press any key to continue . . . _
```

2.Write a C++ Program to accept an alphabet and check whether it is a vowel or a consonant. If it is a vowel, return its predecessor, else its successor. Use call-by-reference with reference arguments.

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

void checkAlphabet(char alphabet, char& result)
{

   if (alphabet == 'a' || alphabet == 'e' || alphabet == 'i' || alphabet == 'o' || alphabet == 'u'||
   alphabet == 'A' || alphabet == 'E' || alphabet == 'I' || alphabet == 'O' || alphabet == 'U')
   {


     result = alphabet - 1; // Return its predecessor
     cout<<alphabet<<" is vowel"<<" its predecessor is "<<result<<endl;
   } else
   {


     result = alphabet + 1; // Return its successor
     cout<<alphabet<<" is consonant"<<" its successor is "<<result<<endl;
   }
}

int main()
{
   char alphabet, result;
   cout << "Enter an alphabet: ";
   cin >> alphabet;
   if(!isalpha(alphabet))
    cout<<"Entered character is not a alphabet"<<endl;
    else
```
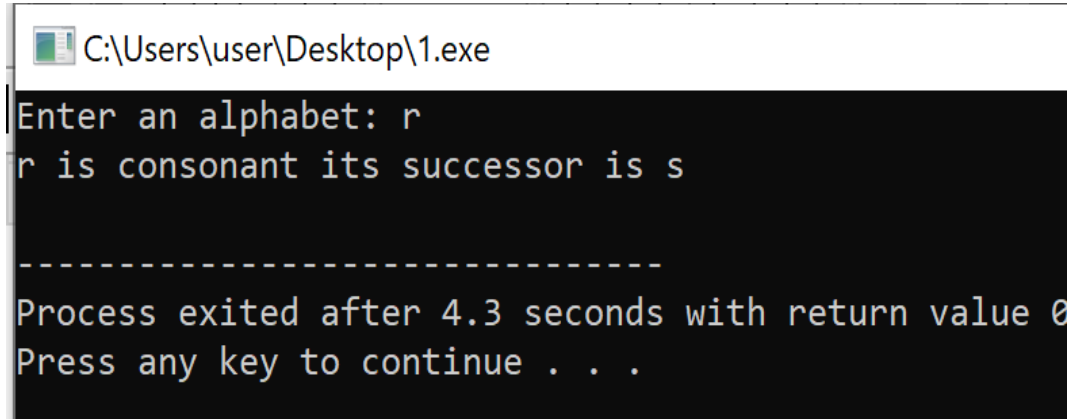
```
    checkAlphabet(alphabet, result);
    return 0;
}
```

OUTPUT

3. Write a C++ Program to call a C function using an extern "C" linkage directive.Use compound statement linkage directive for #include<math.h>.

```
#include <iostream>
#include <math.h>
Using namespace std;

extern "C" {
  double my_sqrt(double x) {
    return sqrt(x);
  }
}

int main() {
  double x = 100;
  double y = my_sqrt(x);
  cout << "Square root of " << x << " is " << y << endl;
  return 0;
```
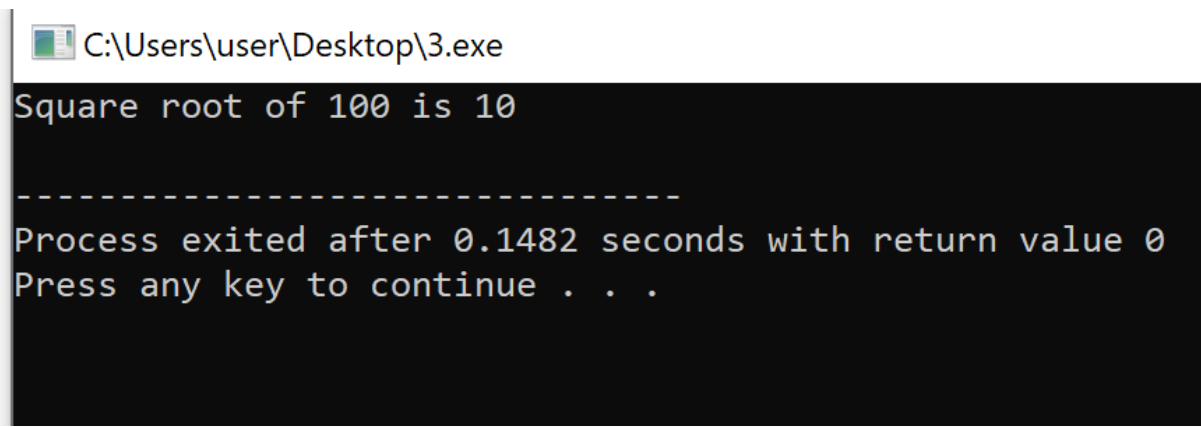


```
C:\Users\user\Desktop\3.exe
Square root of 100 is 10

--------------------------------
Process exited after 0.1482 seconds with return value 0
Press any key to continue . . .
```

4. Write a C++ Program to accept a line of text and count the number of words,characters and digits in it.

```cpp
#include <iostream>
#include <string>
#include <cctype>

using namespace std;

int main()
{
    string line;
    int wordCount = 0, charCount = 0, digitCount = 0;

    cout << "Enter a line of text: ";
    getline(cin, line);

    for (int i = 0; i < line.length(); i++)
    {
        if (isalpha(line[i]))
        {
            charCount++;
        }
        else if (isdigit(line[i]))
        {
            digitCount++;
        }

        if (isspace(line[i]) && !isspace(line[i-1]))
        {
            wordCount++;
        }
    }
}
```

```cpp
    // count the last word if it exists
    if (!isspace(line[line.length()-1]))
    {
        wordCount++;
    }

    cout << "Word count: " << wordCount << endl;
    cout << "Character count: " << charCount << endl;
    cout << "Digit count: " << digitCount << endl;

    return 0;
}
```

OUTPUT

C:\Users\user\Desktop\4.exe

```
Enter a line of text: hellloooo
Word count: 1
Character count: 9
Digit count: 0

------------------------------
Process exited after 3.791 seconds with return value 0
Press any key to continue . . .
```

5. Write a C++ Program to store two binary numbers in arrays and perform bitwise AND, OR and XOR operations on these two numbers.

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

void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void andOperation(int arr1[], int arr2[], int n) {
    int result[n];
    for (int i = 0; i < n; i++) {
        result[i] = arr1[i] & arr2[i];
    }
    cout << "Bitwise AND: ";
    printArray(result, n);
}

void orOperation(int arr1[], int arr2[], int n) {
    int result[n];
    for (int i = 0; i < n; i++) {
        result[i] = arr1[i] | arr2[i];
    }
    cout << "Bitwise OR: ";
    printArray(result, n);
}

void xorOperation(int arr1[], int arr2[], int n) {
    int result[n];
    for (int i = 0; i < n; i++) {
        result[i] = arr1[i] ^ arr2[i];
```

```cpp
        }
        cout << "Bitwise XOR: ";
        printArray(result, n);
    }

    int main() {
        int n;
        cout << "Enter the number of bits: ";
        cin >> n;

        int arr1[n], arr2[n];
        cout << "Enter the first binary number: ";
        for (int i = 0; i < n; i++) {
            cin >> arr1[i];
        }

        cout << "Enter the second binary number: ";
        for (int i = 0; i < n; i++) {
            cin >> arr2[i];
        }
        cout<<"Binary 1=";
        printArray(arr1, n);
        cout<<"Binary 2=";
        printArray(arr2, n);
        andOperation(arr1, arr2, n);
        orOperation(arr1, arr2, n);
        xorOperation(arr1, arr2, n);

        return 0;
    }
```
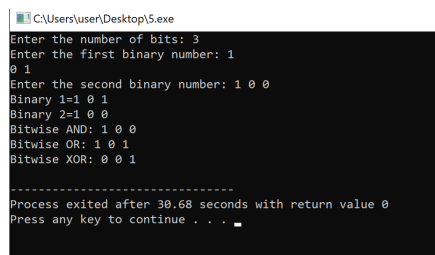


```
C:\Users\user\Desktop\5.exe

Enter the number of bits: 3
Enter the first binary number: 1
0 1
Enter the second binary number: 1 0 0
Binary 1=1 0 1
Binary 2=1 0 0
Bitwise AND: 1 0 0
Bitwise OR: 1 0 1
Bitwise XOR: 0 0 1

------------------------------
Process exited after 30.68 seconds with return value 0
Press any key to continue . . .
```

6.Create two classes D1 and D2 which store the value of distances. D1 stores distances in meters and centimeters and D2 in feet and inches. Write a program to add objects of two classes D1 and D2 and the display the results in feet and inches using friend function.

```cpp
#include <iostream.h>
using namespace std;

class D2;  // forward declaration

class D1 {
  private:
    int meters;
    int centimeters;

  public:
  void get_data()
    {
      cout<<"Enter distance in meters and centimeters: ";
      cin>>meters>>centimeters;
    }
    friend void add(D1 obj1, D2 obj2);
};

class D2 {
  private:
    int feet;
    int inches;

  public:
  void get_data()
    {
      cout<<"Enter distance in feet and inches: ";
      cin>>feet>>inches;
    }
```

```cpp
        friend void add(D1 obj1, D2 obj2);
};

void add(D1 obj1, D2 obj2) {
    // Convert D1 to inches
    int d1_inches = obj1.meters * 39.37 + obj1.centimeters * 0.3937;

    // Add D1 and D2 in inches
    int total_inches = d1_inches + obj2.feet * 12 + obj2.inches;

    // Convert total inches to feet and inches
    int feet = total_inches / 12;
    int inches = total_inches % 12;

    // Display the result in feet and inches
    cout << "Total distance: " << feet << " feet, " << inches << " inches" << endl;
}

int main() {

    D1 d1;
    D2 d2;
    d1.get_data();
    d2.get_data();
    // Add the two objects and display the result
    add(d1, d2);
    return 0;
}
```



```
C:\Users\user\Desktop\ENGINEERING\LAB PROGRAMS\6oops.exe

Enter distance in meters and centimeters: 10 5
Enter distance in feet and inches: 12 7
Total distance: 45 feet, 6 inches

--------------------------------
Process exited after 13.62 seconds with return value 0
Press any key to continue . . .
```

7. Given that an EMPLOYEE class contains the data members like E_Number, E_Name, Basic_salary, DA, HRA, Net_salary and the member functions like Read(), Calculate_Net_Sal(), and Display(). Write a C++ Program to read the data of N Employees and Compute the Net_Salary of each employee.

```cpp
#include <iostream.h>
using namespace std;
class employee
{
  int  emp_number;
  char emp_name[20];
  float emp_basic;
  float emp_da;
  float emp_hra;
  float emp_net_sal;

public:
  void read();
  float calculate_net_salary();
  void display();
};

void employee :: read()
{
  cout << "\nEnter employee number: ";
  cin >> emp_number;
  cout << "\nEnter employee name: ";
  cin >> emp_name;
  cout << "\nEnter employee basic: ";
  cin >> emp_basic;
  cout << "\nEnter employee DA: ";
  cin >> emp_da;
  cout << "\nEnter employee HRA: ";
  cin >> emp_hra;
}
```

```cpp
float employee :: calculate_net_salary()
{
  emp_net_sal = emp_basic + emp_da + emp_hra;
  return emp_net_sal;
}

void employee :: display()
{
  cout << "\n\n**** Details of  Employee ****";
  cout << "\nEmployee Name    : " << emp_name;
  cout << "\nEmployee number   : " << emp_number;
  cout << "\nBasic salary     : " << emp_basic;
  cout << "\nEmployee DA      : " << emp_da;
  cout << "\nEmployee HRA     : " << emp_hra;
  cout << "\nNet Salary      : " << emp_net_sal;
  cout << "\n---------------------------\n\n";
}

int main()
{
  int num_employees;
  int i;
  cout << "How many employees do you want to enter? ";
  cin >> num_employees;
  //employee* emp = new employee[num_employees];
  employee emp[num_employees];

  for ( i = 0; i < num_employees; i++)
  {
    emp[i].read();
    emp[i].calculate_net_salary();
  }

  for ( i = 0; i < num_employees; i++)
```

```
    {
        emp[i].display();
    }
//delete[] emp;
    return 0;
}
```

C:\Users\user\Desktop\ENGINEERING\LAB PROGRAMS\7oops.exe

```
Enter employee DA: 3500

Enter employee HRA: 4000


**** Details of  Employee ****
Employee Name       :   vish
Employee number     :   1
Basic salary        :   10000
Employee DA         :   2000
Employee HRA        :   2500
Net Salary          :   14500
-------------------------------


**** Details of  Employee ****
Employee Name       :   vesh
Employee number     :   2
Basic salary        :   20000
Employee DA         :   3500
Employee HRA        :   4000
Net Salary          :   27500
-------------------------------


-------------------------------
Process exited after 73.83 seconds with return value 0
Press any key to continue . . .
```

8. Write a C++ program to overload the function Search() to search an integer key value and a key value of type double.

```cpp
#include <iostream.h>
using namespace std;

// Function to search an integer key value
int Search(int arr[], int size, int key) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}

// Function to search a key value of type double
int Search(double arr[], int size, double key) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}

int main() {
    int intArr[] = { 10, 20, 30, 40, 50 };
    double doubleArr[] = { 1.1, 2.2, 3.3, 4.4, 5.5 };

    // Searching for an integer key value
    int intIndex = Search(intArr, 5, 30);
    if (intIndex != -1) {
        cout << "Integer key value found at index  " << intIndex << endl;
    }
```

```cpp
        else {
            cout << "Integer key value not found" << endl;
        }

        // Searching for a key value of type double
        int doubleIndex = Search(doubleArr, 5, 3.3);
        if (doubleIndex != -1) {
            cout << "Double key value found at index  " << doubleIndex << endl;
        }
        else {
            cout << "Double key value not found" << endl;
        }

        return 0;
    }
```

```
Integer key value found at index 2
Double key value found at index 2


------------------------------
Process exited after 0.1579 seconds with return value 0
Press any key to continue . . .
```

9. Write a C++ program to find the following using Function Template

a) Successor value of any input of type integer, float, char and double.

b) Sum of all the elements of an array of integers or floats or doubles.

```cpp
#include <iostream.h>
using namespace std;

template <class T>
T successor(T x) {
    return x + 1;
}

template <class X>
X sum(X arr[], int size) {
    X total = 0;
    for (int i = 0; i < size; i++) {
        total += arr[i];
    }
    return total;
}

int main() {
    // Finding the successor value of any input
    cout << "Successor of 5: " << successor(5) << endl;
    cout << "Successor of 5.5: " << successor(5.5) << endl;
    cout << "Successor of 'a': " << successor('a') << endl;

    // Finding the sum of all the elements of an array
    int int_arr[] = {1, 2, 3, 4, 5};
    float float_arr[] = {1.5, 2.5, 3.5, 4.5, 5.5};
    double double_arr[] = {1.0, 2.0, 3.0, 4.0, 5.0};
    int int_arr_size = sizeof(int_arr) / sizeof(int);
    int float_arr_size = sizeof(float_arr) / sizeof(float);
    int double_arr_size = sizeof(double_arr) / sizeof(double);
```
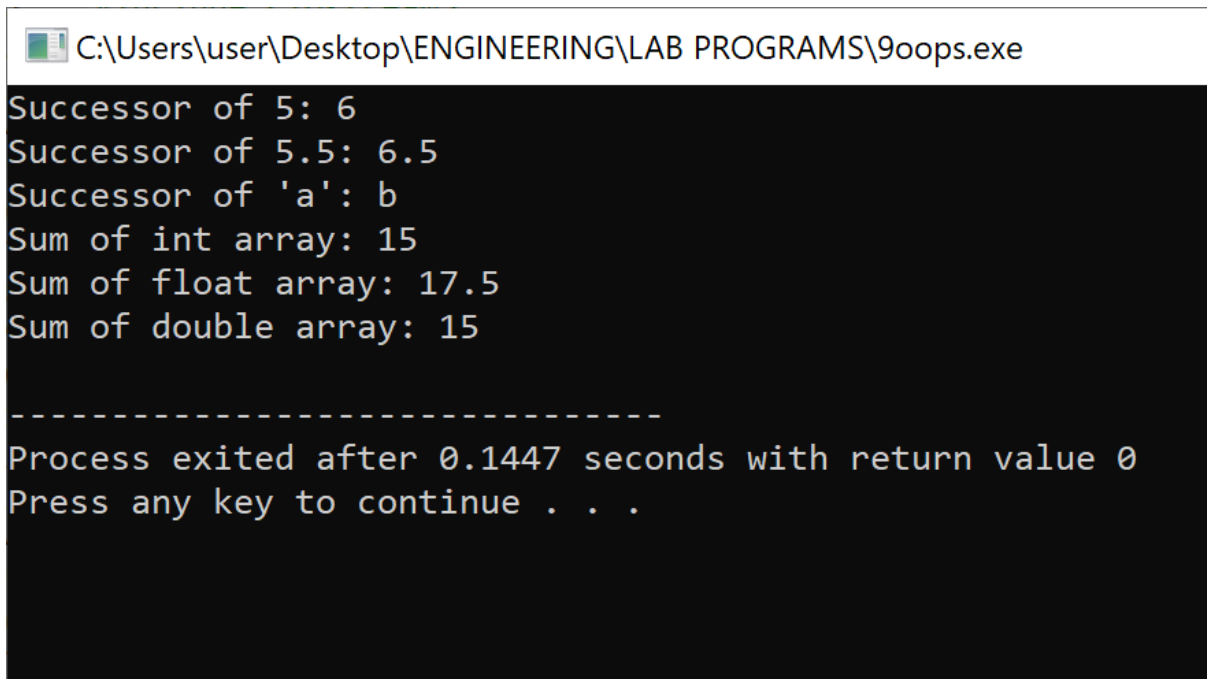
```
    cout << "Sum of int array: " << sum(int_arr, int_arr_size) << endl;
    cout << "Sum of float array: " << sum(float_arr, float_arr_size) << endl;
    cout << "Sum of double array: " << sum(double_arr, double_arr_size) << endl;


    return 0;
}
```

OUTPUT

C:\Users\user\Desktop\ENGINEERING\LAB PROGRAMS\9oops.exe

```
Successor of 5: 6
Successor of 5.5: 6.5
Successor of 'a': b
Sum of int array: 15
Sum of float array: 17.5
Sum of double array: 15


--------------------------------
Process exited after 0.1447 seconds with return value 0
Press any key to continue . . .
```

10.Write a C++ Program to create a class as COMPLEX and implement the following by overloading the function ADD() which returns the Complex numbers
 a) ADD(C1, C2); C1 is an integer ; C2 is a Complex number.
 b) ADD(C1, C2); C1 and C2 are Complex numbers.

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

class COMPLEX {
  private:
    float real;
    float imag;

  public:
    COMPLEX() {  // default constructor
      real = 0;
      imag = 0;
    }

    COMPLEX(float r, float i) {  // parameterized constructor
      real = r;
      imag = i;
    }

    COMPLEX(int r) {  // constructor with integer input
      real = r;
      imag = 0;
    }

    COMPLEX ADD(COMPLEX C) {  // ADD() function for adding two Complex numbers
      COMPLEX res;
      res.real = real + C.real;
      res.imag = imag + C.imag;
      return res;
    }
```

```cpp
    COMPLEX ADD(int r) {  // ADD() function for adding an integer and a Complex number
        COMPLEX res;
        res.real = real + r;
        res.imag = imag;
        return res;
    }

    void display() { // function to display Complex number
        if(imag < 0)
            cout << real << " – i" << abs(imag) << endl;
        else
            cout << real << " + i" << imag << endl;
    }
};

int main() {
    COMPLEX C1(4, 5);
    COMPLEX C2(3, -2);
    COMPLEX C3;

    cout << "C1 = ";
    C1.display();

    cout << "C2 = ";
    C2.display();

    C3 = C1.ADD(C2);  // Adding two Complex numbers
    cout << "C1 + C2 = ";
    C3.display();

    C3 = C1.ADD(2);  // Adding an integer and a Complex number
    cout << "C1 + 2 = ";
    C3.display();
```

```
    return 0;
}
```

```
C1=4+i5
C2=3-i2
C1+C2=7+i3
C1+2=6+i5

--------------------------------
Process exited after 0.1688 seconds with return value 0
Press any key to continue . . .
```

11. Write a C++ Program to create a class called DATE. Accept two valid dates in the form of DD/MM/YYYY. Implement the following by overloading the operators – (minus) and + (Plus). Display the results by overloading the operator << after every operation.

a) No_of _days = D1-D2

b) D1=D1 + No_of _days

 where d1 and d2 are date objects, d1>d2 and No_of_days is an integer.

```cpp
#include<iostream.h>
#include<conio.h>
#include<process.h>
class date
{
int flag,dd,mm,yy;
public:
date(int d,int m,int y)
{
dd=d;mm=m;yy=y;
if((y%4)==0)
flag=1;
else
{
flag=0;
if(m==2 && d>28)
{
cout<<"Error since not a leap year";
getch();
exit(0);
}
}
}
int return_date(date d);
int operator -(date d2);
date operator +(int n);
friend ostream& operator <<(ostream& out,date d1);
```

```
};
int a[20]={0,31,28,31,30,31,30,31,31,30,31,30,31};
int b[20]={0,31,29,31,30,31,30,31,31,30,31,30,31};
int date::operator -(date d2)
{
int a1,a2,x=0;
date d1(dd,mm,yy);
if(d1.dd==d2.dd&&d1.mm==d2.mm&&d1.yy==d2.yy)
return x;
if(d1.dd<d2.dd&&d1.mm==d2.mm&&d1.yy==d2.yy)
{
cout<<"Error since 2nd date is greater than 1st date";
getch();
exit(0);
}
if(d1.mm<d2.mm&&d1.yy==d2.yy)
{
cout<<"Error since 2nd date is greater than 1st date";
getch();
exit(0);
 }
 if(d1.yy<d2.yy)
 {
cout<<"Error since 2nd date is greater than 1st date";
getch();
exit(0);
 }
 a1=return_date(d1);
 a2=return_date(d2);
 for(int i=d1.yy-1;i>d2.yy;i--)
 {
if(i%yy==0)
x=x+366;
else
x=x+365;
```

```
}
if(d1.yy==d2.yy)
x=x+a1-a2;
else
{
x=x+a1;
if(d2.yy%4==0)
x=x+(366-a2);
else
x=x+(365-a2);
}
return x;
}
int date::return_date(date d)
{
int int_val=d.dd;
if((d.flag==1)&&(d.mm>2))
{
for(int i=0;i<d.mm;i++)
int_val=int_val+a[i];
int_val++;
}
else
{
for(int i=0;i<d.mm;i++)
int_val=int_val+a[i];
}
return int_val;
}
date date::operator +(int n)
{
date d(dd,mm,yy);
for(int i=0;i<n;i++)
{
d.dd++;
```

```cpp
if(d.yy%4==0)
{
d.flag=1;
if(d.dd>b[d.mm])
{
d.dd=1;
d.mm++;
}
}
else
{
d.flag=0;
if(d.dd>a[d.mm])
{
d.dd=1;
d.mm++;
}
}
if(d.mm>12)
{
d.mm=1;
d.yy++;
}
}
return d;
}
ostream& operator <<(ostream& out,date d1)
{
out << d1.dd<<"-"<<d1.mm<<"-"<<d1.yy;
return out;
}
int main()
{
int d,m,y,no_of_days;
clrscr();
```

```cpp
cout<<"Enter a valid date(dd mm yyyy):\n";
cin>>d>>m>>y;
if(d>b[m]||m>12)
{
cout<<"Error";
getch();
exit(0);
}
date d1(d,m,y);
cout<<"Enter a valid date which is lesser than the first date\n";
cin>>d>>m>>y;
if(d>b[m]||m>12)
{
cout<<"Error";
getch();
exit(0);
}
date d2(d,m,y);
no_of_days=d1-d2;
cout<<"Difference b/w two dates in days :"<<no_of_days<<endl;
cout<<"Enter a no of days to be added to first date:\n";
cin>>no_of_days;
d1=d1+no_of_days;
cout<<"The new date is:"<<d1<<endl;
getch();
return 0;
}
```



```
C:\Users\user\Desktop\ENGINEERING\LAB PROGRAMS\11oops.exe

Enter a valid date(dd mm yyyy):
11 09 2014
Enter a valid date which is lesser than the first date
11 09 2013
Difference b/w two dates in days :365
Enter a no of days to be added to first date:
35
The new date is:16-10-2014
```

12. Write a C++ program to create a class called MATRIX using a two-dimensional array of integers. Implement the following by overloading the operator == which checks the compatibility of two matrices to be added and subtracted. Perform the following by overloading + and - operators. Display the result by overloading the operator <<.

```cpp
#include<iostream>
using namespace std;
class matrix
{
 private:long m[5][5];
 int row;int col;
 public:void getdata();
 int operator ==(matrix);
 matrix operator+(matrix);
 matrix operator-(matrix);
 friend ostream & operator << (ostream &,matrix &);
};
/* function to check whether the order of matrix are same or not */
int matrix::operator==(matrix cm)
{
 if(row==cm.row && col==cm.col)
 {
   return 1;
 }
 return 0;
}
/* function to read data for matrix*/
void matrix::getdata()
{
 cout<<"enter the number of rows\n";
 cin>>row;
 cout<<"enter the number of columns\n";
 cin>>col;
 cout<<"enter the elements of the matrix\n";
```

```cpp
  for(int i=0;i<row;i++)
  {
   for(int j=0;j<col;j++)
   {
     cin>>m[i][j];
   }
  }
}
/* function to add two matrix */
matrix matrix::operator+(matrix am)
{
  matrix temp;
  for(int i=0;i<row;i++)
  {
   for(int j=0;j<col;j++)
   {
     temp.m[i][j]=m[i][j]+am.m[i][j];
   }
   temp.row=row;
   temp.col=col;
  }
  return temp;
}
/* function to subtract two matrix */
matrix matrix::operator-(matrix sm)
{
  matrix temp;
  for(int i=0;i<row;i++)
  {
   for(int j=0;j<col;j++)
   {
     temp.m[i][j]=m[i][j]-sm.m[i][j];
   }
   temp.row=row;
   temp.col=col;
```

```cpp
 }
 return temp;
}
/* function to display the contents of the matrix */
ostream & operator <<(ostream &fout,matrix &d)
{
 for(int i=0;i<d.col;i++)
 {
  for(int j=0;j<d.col;j++)
  {
   fout<<d.m[i][j];
   cout<<" ";
  }
  cout<<endl;
 }
return fout;
}
/* main function */
int main()
{
 matrix m1,m2,m3,m4;

 m1.getdata();
 m2.getdata();
 if(m1==m2)
 {
  m3=m1+m2;
  m4=m1-m2;
  cout<<"Addition of matrices\n";
  cout<<"the result is\n";
  cout<<m3;
  cout<<"subtraction of matrices\n";
  cout<<"The result is \n";
  cout<<m4;
 }
```

```cpp
else
{
  cout<<"order of the input matrices is not identical\n";
}


}
```

```
enter the number of rows
2
enter the number of columns
2
enter the elements of the matrix
1 2 3 4
enter the number of rows
2
enter the number of columns
2
enter the elements of the matrix
3 4 5 7
Addition of matrices
the result is
4 6
8 11
subtraction of matrices
The result is
-2 -2
-2 -3
```

13. Write a C++ program to create a class called STUDENT with data members USN,Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester,Fees and Stipend. Enter the data for at least 5 students. Find the semester wise average age for all UG and PG students separately.

```cpp
#include<iostream>
using namespace std;
class student
{
 public:  int reg,age;
       char name[20];
       void read_data();
};

class ugstudent:public student
{
 public:  int stipend,sem;
        float fees;
       void read_data();
};

class pgstudent:public student
{
 public:  int stipend,sem;
        float fees;
       void read_data();
};

/* function to read student details*/
void student::read_data()
{
 cout<<"\n Enter name:";
 cin>>name;
```

```cpp
cout<<"\n Enter Reg.no.";
cin>>reg;
cout<<"\n Enter age:";
cin>>age;
}
void ugstudent::read_data()
{
student::read_data();
cout<<"\nEnter the sem:";
cin>>sem;
cout<<"\nEnter the fees:";
cin>>fees;
cout<<"\nEnter the stipend:";
cin>>stipend;
}

/* function to read additional details for pgstudents*/
void pgstudent::read_data()
{
student::read_data();
cout<<"\nEnter the sem:";
cin>>sem;
cout<<"\nEnter the fees:";
cin>>fees;
cout<<"\nEnter the stipend:";
cin>>stipend;
}

/* main function */
int main()
{
ugstudent ug[20];
pgstudent pg[20];
int i,n,m;
float average;
```

```cpp
cout<<"\nEnter the no. of entries in the ugstudent class:";
cin>>n;
for(i=1;i<=n;i++)
ug[i].read_data();
for(int sem=1;sem<=8;sem++)
{
float sum=0;
int found=0,count=0;
for(i=1;i<=n;i++)
{
if(ug[i].sem==sem)
{
sum=sum+ug[i].age;
found=1;
 count++;
}
}
if(found==1)
{
average=sum/count;
cout<<"\nAverage of age of sem "<<sem<<" is "<<average;

}
}
cout<<"\nEnter the no. of entries of pgstudent class:";
cin>>n;
for(i=1;i<=n;i++)
pg[i].read_data();
for(sem=1;sem<=8;sem++)
{
float sum=0;
int found=0,count=0;
for(i=1;i<=n;i++)
{
if(pg[i].sem==sem)
```

```
{
sum=sum+pg[i].age;
found=1;
count++;
}
}
if(found==1)
{
average=sum/count;
cout<<"\nAverage of age of sem "<<sem<<" is "<<average;
}
}


}
```

```
  Enter age:30
 Enter the sem:2
 Enter the fees:3000000
 Enter the stipend:23000
 Average of age of sem 2 is 30
 Enter the no. of entries of pgstudent class:1
  Enter name:dhoni
  Enter Reg.no.001
  Enter age:40
 Enter the sem:2
 Enter the fees:300000
 Enter the stipend:2300
 Average of age of sem 2 is 40_
```

14. Write a C++ program to create a base class Student, from this inherit a new class called Exams, containing Marks1, Marks2 and Marks3 as its data members. Also create another class called Sports having Sports_grade as its data member. Now, create yet another class but a derived class of Exams and Sports classes and call it as Awards. Use appropriate member functions in all classes. You can use the Regno, Name, Semester and Branch as the members of the base class.

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

class Student {
  public:
    int Regno;
    char Name[20];
    int Semester;
    char Branch[20];
    void GetData() {
      cout << "Enter Regno: ";
      cin >> Regno;
      cout << "Enter Name: ";
      cin >> Name;
      cout << "Enter Semester: ";
      cin >> Semester;
      cout << "Enter Branch: ";
      cin >> Branch;
    }
};

class Exams : public virtual Student {
  public:
    int Marks1;
    int Marks2;
    int Marks3;
    void GetMarks() {
      cout << "Enter Marks1: ";
```

```cpp
        cin >> Marks1;
        cout << "Enter Marks2: ";
        cin >> Marks2;
        cout << "Enter Marks3: ";
        cin >> Marks3;
    }
};


class Sports : public virtual Student {
  public:
    char Sports_grade;
    void GetSports() {
      cout << "Enter Sports Grade: ";
      cin >> Sports_grade;
    }
};


class Awards : public Exams, public Sports {
  public:
    void Display() {
      cout << "Name: " << Student::Name << endl;
      cout << "Regno: " << Student::Regno << endl;
      cout << "Branch: " << Student::Branch << endl;
      cout << "Exams Semester: " << Exams::Semester << endl;
      cout << "Marks1: " << Marks1 << endl;
      cout << "Marks2: " << Marks2 << endl;
      cout << "Marks3: " << Marks3 << endl;
      cout << "Sports Semester: " << Sports::Semester << endl;
      cout << "Sports Grade: " << Sports_grade << endl;
    }
};


int main() {
  Awards a1;
  a1.GetData();
```

```
    a1.GetMarks();

    a1.GetSports();

    a1.Display();


    return 0;
}
```

C:\Users\user\Desktop\ENGINEERING\LAB PROGRAMS\14oops.exe

```
Enter Regno: 225
Enter Name: hillrush
Enter Semester: 4
Enter Branch: mining
Enter Marks1: 98
Enter Marks2: 97
Enter Marks3: 99
Enter Sports Grade: A
Name: hillrush
Regno: 225
Branch: mining
Exams Semester: 4
Marks1: 98
Marks2: 97
Marks3: 99
Sports Semester: 4
Sports Grade: A
```

15. Write a C++ program to create a file called File1.txt and print its contents in the reverse order to the File2.txt.

```cpp
#include <iostream>
#include<algorithm>
#include<fstream>

using namespace std;


int main(){
   string str,st;
   ifstream in("sample.txt");
   ofstream out("sample2.txt");

   while (!in.eof())
   {
      getline(in,st);
      str= str + st;
      str+="\n";
   }
   in.close();
   reverse(str.begin(),str.end()-1);

   out<<str<<endl;
   out.close();

   return 0;
}
```

sample - Notepad
File Edit Format View Help
welcome

sample2 - Notepad
File Edit Format View Help
emoclew