

Name: Priyanshu Sharma

Roll No: CS24042

Practical 1

Program to demonstrate use of data members & member functions.

Write a C++ Program which computes area and perimeter of a rectangle. The program should take length and breadth of the rectangle as input, calculates the area and perimeter of the rectangle and output it on the screen.

Code:

```
#include <iostream>

using namespace std;

class Rectangle
{
public:
    int length;
    int breadth
    int area()
    {
        return length * breadth;
    }
    int perimeter()
    {
        return 2 * (length + breadth);
    }
};

int main()
{
```

Name: Priyanshu Sharma

Roll No: CS24042

Rectangle rect;

cout << "Length of rectangle: ";

cin >> rect.length;

cout << "\nBreadth of rectangle: ";

cin >> rect.breadth;

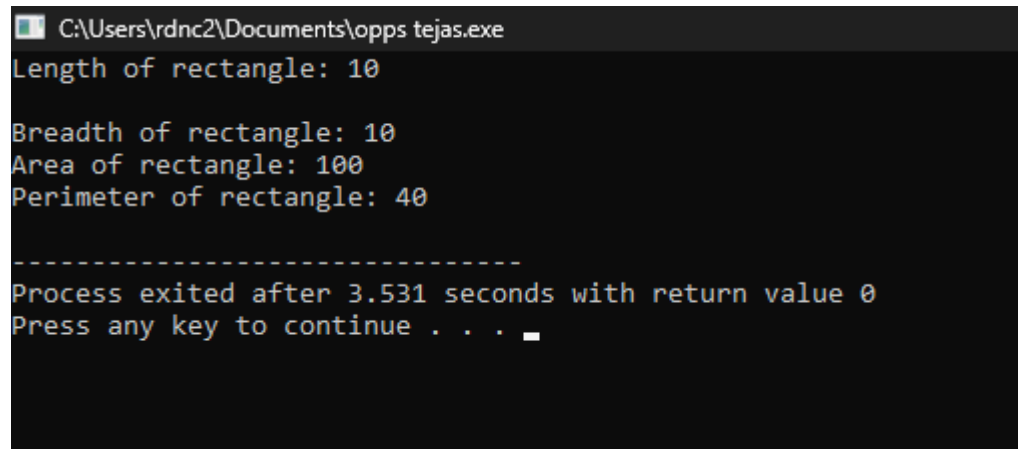
cout << "Area of rectangle: " << rect.area() << endl;

cout << "Perimeter of rectangle: " << rect.perimeter() << endl;

return 0;

}

output



```
C:\Users\rdnc2\Documents\opps tejas.exe
Length of rectangle: 10

Breadth of rectangle: 10
Area of rectangle: 100
Perimeter of rectangle: 40

-----
Process exited after 3.531 seconds with return value 0
Press any key to continue . . . _
```

Name: Priyanshu Sharma

Roll No: CS24042

Practical 2

Programs based on branching and looping statements using classes.

Write a C++ Program which computes grade of the students in the exam. The program should take total marks as an input, calculates the grade and output it on the screen. For Grade calculation follow the following chart -

Marks Grade

<40 F

>=40 and <50 E

>=50 and <60 D

>=60 and <65 C

>=65 and <70 B

>=70 and <75 B+

>=75 and < 80 A

>=80 and <85 A+

>= 85 and <90 A++

>=90 O

Code:

#include <iostream>

using namespace std;

class Student

{

public:

int marks;

Name: Priyanshu Sharma

Roll No: CS24042

void getMarks()

{

cout << "Enter marks: ";

cin >> marks;

}

void calculateGrade()

{

if (marks >= 90) {

cout << "Grade: O – Outstanding " << endl;

}

else

if (marks >= 85 and marks < 90) {

cout << "Grade: A++" << endl;

}

else

if (marks >= 80 and marks < 85) {

cout << "Grade: A+" << endl;

}

else

if (marks >= 75 and marks < 80) {

cout << "Grade: A" << endl;

}

else

if (marks >= 70 and marks < 75) {

Name: Priyanshu Sharma

Roll No: CS24042

cout << "Grade: B+" << endl;

}

else

if (marks >= 70 and marks < 75) {

cout << "Grade: B+" << endl;

}

else

if (marks >= 65 and marks < 70) {

cout << "Grade: B" << endl;

}

else

if (marks >= 60 and marks < 65) {

cout << "Grade: C" << endl;

}

else

if (marks >= 50 and marks < 60) {

cout << "Grade: D" << endl;

}

else

if (marks >= 40 and marks < 50) {

cout << "Grade: E" << endl;

}

else

{

Name: Priyanshu Sharma

Roll No: CS24042

cout << "Grade: F" << endl;

}

}

};

int main()

Student student;

student.getMarks();

student.calculateGrade();

return 0;

}

OUTPUT

```
Enter marks: 90
Grade: 0 Outstanding
-----
Process exited after 2.388 seconds with return value 0
Press any key to continue . . . |
```

Name: Priyanshu Sharma

Roll No: CS24042

Practical 3:

Program to demonstrate one and two dimensional arrays using classes.

3.1

Write a C++ Program which calculates sum of 'n' natural numbers. The program should take number as an input in an array, calculates the sum and output it on the screen.

Code:

#include <iostream>

using namespace std;

int main()

{

int arr[100], i, size, sum=0, n;

cout<<"Enter the number of elements: ";

cin>>size; //Accepting array size

cout<<"Enter the value of elements: "<<endl;

n = size;

for(i=0; i<n; i++)

{

cin>>arr[i]; //Accepting values

}

for(i=0; i<n; i++)

{

sum=sum+arr[i]; //Calculating sum

}

cout<<"Sum of elements in an array is: "<<sum;

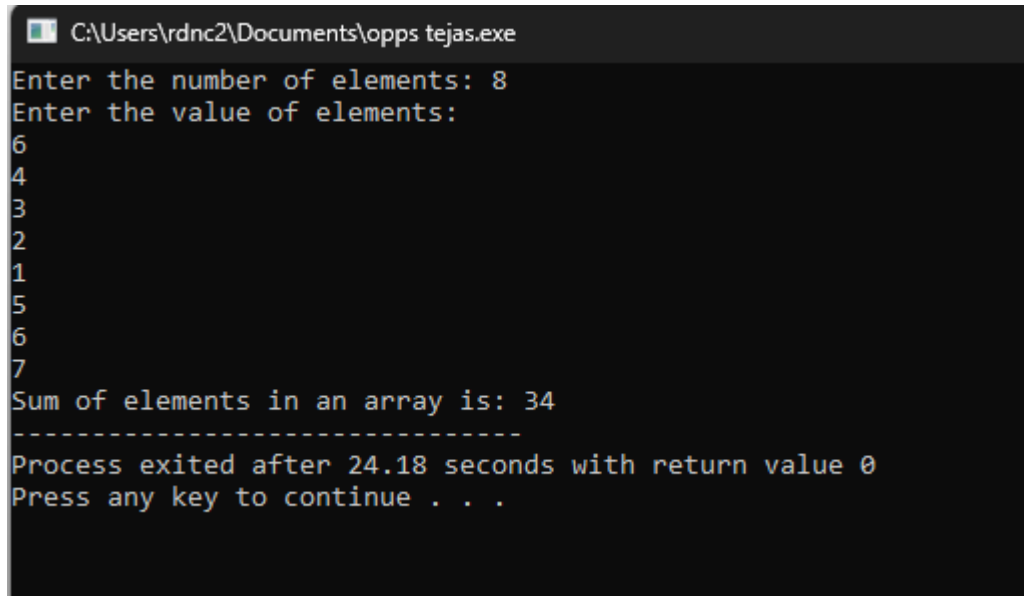
return 0;

Name: Priyanshu Sharma

Roll No: CS24042

}

Output:



```
C:\Users\rdnc2\Documents\opps tejas.exe
Enter the number of elements: 8
Enter the value of elements:
6
4
3
2
1
5
6
7
Sum of elements in an array is: 34
-----
Process exited after 24.18 seconds with return value 0
Press any key to continue . . .
```

3.2

Write a C++ Program which calculates sum of two matrix. The program should take input for matrix 1 and 2 using 2 dimensional array, calculates the sum and output it on the screen.

Code:

#include<iostream>

using namespace std;

main()

{

int m1[5][5], m2[5][5], m3[5][5];

int i, j, r, c;

cout<<"Enter the no.of rows of the matrices to be added(max 5):";

cin>>r;

cout<<"Enter the no.of columns of the matrices to be added(max 5):";

Name: Priyanshu Sharma

Roll No: CS24042

cin>>c;

cout<<"\n1st Matrix Input:\n";

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<"\nmatrix1["<<i<<"]["<<j<<"]=";

cin>>m1[i][j];

}

}

cout<<"\n2nd Matrix Input:\n";

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<"\nmatrix2["<<i<<"]["<<j<<"]=";

cin>>m2[i][j];

}

}

cout<<"\nAdding Matrices...\n";

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

m3[i][j]=m1[i][j]+m2[i][j];

}

Name: Priyanshu Sharma

Roll No: CS24042

```
}  
  
cout<<"\nThe resultant Matrix is:\n";  
  
for(i=0;i<r;i++)  
{  
    for(j=0;j<c;j++)  
    {  
        cout<<"\t"<<m3[i][j];  
    }  
    cout<<endl;  
}  
}
```

Output:

```
Enter the no.of rows of the matrices to be added(max 5):2  
Enter the no.of columns of the matrices to be added(max 5):1  
  
1st Matrix Input:  
matrix1[0][0]= 4  
matrix1[1][0]= 6  
  
2nd Matrix Input:  
matrix2[0][0]= 8  
matrix2[1][0]= 2  
  
Adding Matrices...  
  
The resultant Matrix is:  
    12  
    8  
  
-----  
Process exited after 17.08 seconds with return value 0  
Press any key to continue . . . |
```

Name: Priyanshu Sharma

Roll No: CS24042

Practical 4:

Program to use scope resolution operator. Display the various values of the same variables declared at different scope levels declared at different scope levels.

Code:

```
#include <iostream>  
  
using namespace std;  
  
int globalVariable = 10;  
  
void localScopeFunction() {  
  
int localVariable = 5;  
  
cout << "Inside localScopeFunction - Global Variable: "  
<< ::globalVariable << endl;  
  
cout << "Inside localScopeFunction - Local Variable: "  
<< localVariable << endl;  
  
}  
  
int main() {  
  
cout << "Inside main - Global Variable: " <<  
globalVariable << endl;  
  
localScopeFunction();  
  
{  
  
int nestedVariable = 20;  
  
cout << "Inside nested block - Global Variable: " <<  
::globalVariable << endl;  
  
cout << "Inside nested block - Nested Variable: " <<  
nestedVariable << endl;  
  
}
```

Name: Priyanshu Sharma

Roll No: CS24042

return 0;

}

Output:

```
Inside main - Global Variable: 10
Inside localScopeFunction - Global Variable: 10
Inside localScopeFunction - Local Variable: 5
Inside nested block - Global Variable: 10
Inside nested block - Nested Variable: 20

-----
Process exited after 0.2264 seconds with return value 0
Press any key to continue . . . |
```

Name: Priyanshu Sharma

Roll No: CS24042

Practical 5:

Programs to demonstrate various types of constructors and destructors.

Code:

#include <iostream>

using namespace std;

class MyClass {

public:

MyClass() {

cout << "Default Constructor called" << endl;

}

MyClass(int value) {

this->value = value;

cout << "Parameterized Constructor called with value: " << value << endl;

}

MyClass(const MyClass &other) {

this->value = other.value;

cout << "Copy Constructor called with value: " << value << endl;

}

MyClass() {

cout << "Destructor called for value: " << value << endl;

}

private:

int value;

};

int main() {

MyClass obj1;

Name: Priyanshu Sharma

Roll No: CS24042

```
MyClass obj2(42);  
MyClass obj3 = obj2;  
MyClass *dynamicObj1 = new MyClass;  
MyClass *dynamicObj2 = new MyClass(99);  
delete dynamicObj1;  
delete dynamicObj2;  
return 0;  
}
```

Output:

```
Default Constructor called  
Parameterized Constructor called with value: 42  
Copy Constructor called with value: 42  
Default Constructor called  
Parameterized Constructor called with value: 99  
Destructor called for value: 12392992  
Destructor called for value: 99  
Destructor called for value: 42  
Destructor called for value: 42  
Destructor called for value: 1  
  
-----  
Process exited after 1.459 seconds with return value 0  
Press any key to continue . . . |
```

NAME:- Alimehdi Suleman

Roll No: CS24048

Practical 6

Programs to demonstrate use of public, protected & private scope specifiers.

Code:

```
#include <iostream>

using namespace std;

class BaseClass
{
public:
    int publicVar; void
    publicFunction(void)
    {
        cout << "Inside public Function of Base Class" << endl;
    }

protected:
    int protectedVar; void
    protectedFunction(void)
    {
        cout << "Inside protected Function of Base Class" << endl;
    }

private:
    int privateVar; void
    privateFunction()
    {
        cout << "Inside private Function of Base Class" << endl;
    }
}
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
}  
};  
  
int main()  
{  
    BaseClass obj;  
  
    obj.publicVar = 5;  
  
    //obj.privateVar=10;  //obj.protectedVar=15;  
  
    cout <<"Public variable value: "<<  
  
    obj.publicVar<<endl;  obj.publicFunction();  
  
    //obj.protectedFunction();  
  
    //obj.privateFunction();  
  
    return 0;  
}
```

Output:-

```
Public variable value: 5  
Inside public Function of Base Class  
-----  
Process exited after 0.04341 seconds with return value 0  
Press any key to continue . . .
```


NAME:- Alimehdi Suleman
Roll No: CS24048

Practical 7

Programs to demonstrate single and multilevel inheritance

7.1: Single Level Inheritance

Code:

```
#include <iostream>

using namespace std;

class Account
{
    public:
    float salary = 60000;
};

class Programmer: public Account
{
    public:
    float bonus = 5000;
};

int
main(void)
{
    Programmer p1;

    cout<<"Salary:
    "<<p1.salary<<endl;

    cout<<"Bonus:
    "<<p1.bonus<<endl;    return 0;
}
```

NAME:- Alimehdi Suleman
Roll No: CS24048

Output:-

```
Salary: 60000  
Bonus: 5000  
-----  
Process exited after 0.2029 seconds with return value 0  
Press any key to continue . . . _
```

NAME:- Alimehdi Suleman

Roll No: CS24048

7.2: Multi Level Inheritance

Code:

```
// C++ program to implement Multilevel
Inheritance using namespace std; // base class
class Vehicle{include <iostream>

{
public:
Vehicle()
{
cout << "This is a Vehicle" << endl;
}
};

class fourWheeler: public Vehicle
{ public:
fourWheeler()
{
cout<<"Objects with 4 wheels are vehicles"<< endl;
}
};

// sub class derived from two base
classes class Car: public fourWheeler{
public: Car()
{
cout<<"Car has 4 Wheels"<<endl;
}
```

NAME:- Alimehdi Suleman

Roll No: CS24048

};

// main function

int main()

{

//creating object of sub class will

//invoke the constructor of base

classes Car obj; return 0;

}

Output:-

```
This is a Vehicle
Objects with 4 wheels are vehicles
Car has 4 Wheels

-----
Process exited after 0.07261 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman

Roll No: CS24048

7.3: Multi Level Inheritance

Code:

```
#include <iostream>

using namespace std;

class Animal
{
    public: void eat() {
        cout<<"Eating...base
        class"<<endl;
    } }; class Dog:
        public Animal
    { public: void bark() {
        cout<<"Barking...first level derived
        class"<<endl;
    } }; class BabyDog:
        public Dog
    { public:
        void weep()
        { cout<<"Weeping...second level derived
        class";
        }
    };

int main(void)
{
    BabyDog d1;
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
d1.eat();
```

```
d1.bark();
```

```
d1.weep();
```

```
return 0;
```

```
}
```

Output:-

```
Eating...base class
Barking...first level derived class
Weeping...second level derived class
-----
Process exited after 0.04676 seconds with return value 0
Press any key to continue . . . _
```

NAME:- Alimehdi Suleman
Roll No: CS24048

Practical 8

Programs to demonstrate multiple inheritance and hierarchical inheritance.

8.1: Multiple Inheritance

Code:

```
#include <iostream>

using namespace std;

class A
{
protected:

int a;

public:

void get_a(int n)
{
a
= n;
}
};

class B
{
protected:

int b;

public:

void get_b(int n)
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
{    b

= n;

}

};

class C : public A,public B
{

public:

    void display()

    {    cout << "The value of a is : "

<<a<< endl;    cout << "The value of b

is : " <<b<< endl;    cout<<"Addition

of a and b is : "<<a+b;

    }

};

int main()

{

    C c;

    c.get_a(10);

    c.get_b(20);

    c.display();

return 0;

}
```

Output:-

```
The value of a is : 10
The value of b is : 20
Addition of a and b is : 30
-----
Process exited after 0.04317 seconds with return value 0
Press any key to continue . . . _
```


NAME:- Alimehdi Suleman
Roll No: CS24048

8.2: Hierarchical Inheritance

Code:

```
#include <iostream> using
namespace std; class Shape //
Declaration of base class.
{
    public:    int a;    int b;

    void get_data(int n,int m)
    {
a=n;
b = m;
    }
};

class Rectangle : public Shape // inheriting Shape class
{
    public:

    int rect_area()
    {    int result =
a*b;    return
result;
    }
};

class Triangle : public Shape // inheriting Shape class
{
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
public:

    int triangle_area()
    {   float result =
0.5*a*b;   return
result;

    }
};

int main()
{
    Rectangle r;   Triangle t;   int
length,breadth,base,height;   cout << "Enter the
length of a rectangle: " << endl;   cin>>length;
cout << "Enter the breadth of a rectangle: " <<
endl;   cin>>breadth;

    r.get_data(length,breadth);   int m =
r.rect_area();   cout << "Area of the rectangle is
:" <<m<<endl;   cout << "Enter the base of the
triangle: " << endl;   cin>>base;   cout <<
"Enter the height of the triangle: " << endl;
cin>>height;

    t.get_data(base,height);   float n =
t.triangle_area();   cout <<"Area of the
triangle is : " << n<<endl;   return 0;
}
```

Output:-

NAME:- Alimehdi Suleman

Roll No: CS24048

```
Enter the length of a rectangle:
15
Enter the breadth of a rectangle:
10
Area of the rectangle is : 150
Enter the base of the triangle:
5
Enter the height of the triangle:
3
Area of the triangle is : 7

-----
Process exited after 15.29 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman

Roll No: CS24048

8.3: Hybrid Inheritance

Code:

```
// C++ program for Hybrid Inheritance
```

```
#include
```

```
<iostream> using
```

```
namespace std; //
```

```
base class class
```

```
Vehicle { public:
```

```
Vehicle()
```

```
{
```

```
cout << "This is a Vehicle" << endl;
```

```
}
```

```
};
```

```
//base class
```

```
class Fare{
```

```
public:
```

```
Fare()
```

```
{
```

```
cout<<"Fare of Vehicle¥n";
```

```
}
```

```
};
```

```
// first sub class class
```

```
Car: public Vehicle
```

```
{
```

```
};
```

NAME:- Alimehdi Suleman

Roll No: CS24048

// second sub class class Bus:

public Vehicle, public Fare {

};

// main function

int main()

{

// creating object of sub class will invoke the constructor of base class

Bus obj2;

Car obj1;

return 0;

}

Output:-

```
This is a Vehicle
Fare of Vehicle
This is a Vehicle

-----
Process exited after 0.04396 seconds with return value 0
Press any key to continue . . . _
```

NAME:- Alimehdi Suleman
Roll No: CS24048

8.4: Hybrid Inheritance

Code:

```
#include
<iostream> using
namespace std;
class A
{
    protected: int a; public: void
    get_a() { cout << "Enter the value of
    'a' : " << endl; cin>>a;
    }
};
class B : public A { protected: int b;
public: void get_b() { cout <<
"Enter the value of 'b' : " << endl;
cin>>b;
    }
};
class C
{
    protected:
    int c;
    public:
    void get_c()
    {
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
    cout << "Enter the value of c is : " <<
endl;    cin>>c;    }

};

class D : public B, public C
{
    protected:    int d;    public:    void mul()    {
get_a();    get_b();    get_c(); cout <<
"Multiplication of a,b,c is : " <<a*b*c<< endl; }

};

int main()
{
    D d;

    d.mul();

    return 0;

}
```

Output:-

```
Enter the value of 'a' :
10
Enter the value of 'b' :
20
Enter the value of c is :
30
Multiplication of a,b,c is : 6000

-----
Process exited after 14.58 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman

Roll No: CS24048

Practical 9

Programs to demonstrate inheritance and derived class constructors.

Code:

```
#include
<iostream> using
namespace std;
class Base { public:
int x;
Base(int a) : x(a) { cout << "Base class
constructor invoked." << endl;
}
};
class Derived : public
Base { public: int y;
Derived(int a, int b) : Base(a), y(b) { cout <<
"Derived class constructor invoked." <<
endl;
}
void display() { cout << "x = " << x <<
", y = " << y << endl;
}
};
int main() {
Derived d(10,
20);
```


NAME:- Alimehdi Suleman

Roll No: CS24048

d.display();

return 0;

}

Output:-

```
Base class constructor invoked.  
Derived class constructor invoked.  
x = 10, y = 20  
-----  
Process exited after 0.1974 seconds with return value 0  
Press any key to continue . . .
```

NAME:- Alimehdi Suleman

Roll No: CS24048

Practical 10

Programs to demonstrate friend function.

10.1: Friend Function in one Class

Code:

```
// C++ program to demonstrate the working of friend function
```

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

```
class Distance { private: int meter;
```

```
friend int addFive(Distance); // friend
```

```
function public:
```

```
    Distance() : meter(0) {}
```

```
};
```

```
// friend function definition
```

```
int addFive(Distance d) {
```

```
    //accessing private members from the friend function
```

```
    d.meter += 5;
```

```
return d.meter;
```

```
}
```

```
int main() { Distance D; cout
```

```
<< "Distance: " << addFive(D);
```

```
return 0;
```

```
}
```

Output:-

```
Distance: 5
-----
Process exited after 0.2166 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman
Roll No: CS24048

10.2: Friend Function in one Class

Code:

```
#include <iostream>

using namespace std;

class Box
{
    private:
    int length;

    public:
    Box(): length(0) { }    friend int
    printLength(Box); //friend function
};

int printLength(Box b)
{
    b.length += 10;
    return b.length;
}

int main()
{
    Box b;    cout<<"Length of box: "<<
    printLength(b)<<endl;    return 0;
}
```

Output:-

```
Length of box: 10
-----
Process exited after 0.1522 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman

Roll No: CS24048

10.3: Friend Function in two Classes

Code:

```
//two different classes using friend functions

#include <iostream> using
namespace std; class ClassB; //
forward declaration class ClassA
{   public:
    // constructor to initialize numA to 12   ClassA() :
numA(12) {}   private:   int numA;   friend int
add(ClassA, ClassB); // friend function declaration
};

class ClassB {
public:
ClassB() : numB(1) {} // constructor to initialize numB to
1 private:   int numB;   friend int add(ClassA, ClassB);
// friend function declaration
};

// access members of both classes int
add(ClassA objectA, ClassB objectB) {
return (objectA.numA +
objectB.numB);
}

int main() {
    ClassA objectA;
    ClassB objectB;   cout << "Sum: " <<
add(objectA, objectB);   return 0;
```

NAME:- Alimehdi Suleman

Roll No: CS24048

}

Output:-

```
Sum: 13
-----
Process exited after 0.09897 seconds with return value 0
Press any key to continue . . . _
```

NAME:- Alimehdi Suleman
Roll No: CS24048

10.4: Friend Class

Code:

```
#include <iostream>

using namespace std;

class A
{
    int x =5;    friend class B;
    // friend class.
};

class B
{
    public:
    void display(A &a)
    {    cout<<"value of x is :
"<<a.x;
    }
};

int main()
{
    A a;
    B b;
    b.display(a);
    return 0;
}
```

Output:-

NAME:- Alimehdi Suleman
Roll No: CS24048

Practical 11

Programs to demonstrate function overloading and overriding.

11.1: Function Overloading

Code:

```
#include
<iostream> using
namespace std;
class Temp
{
    private:  int x = 10;  double x1 = 10.1;
    public:  void add(int y)  {  cout <<
"Value of x + y is: " << x + y << endl;
    }
    // Differ in the type of argument.
    void add(double d)
    {  cout << "Value of x1 + d is: " << x1 + d
<< endl;
    }
    // Differ in the number of arguments.
    void add(int y, int z)
    {  cout << "Value of x + y + z is: " << x + y + z
<< endl;
    }
};

int main()
```

```
value of x is : 5
```

```
Process exited after 0.05454 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
{  
    Temp t1;  
    t1.add(10);  
    t1.add(11.1);  
    t1.add(12,13);  
    return 0;  
}
```

Output:-

```
Value of x + y is: 20  
Value of x1 + d is: 21.2  
Value of x + y + z is: 35  
-----  
Process exited after 0.04552 seconds with return value 0  
Press any key to continue . . . _
```


NAME:- Alimehdi Suleman
Roll No: CS24048

11.2: Operator Overloading

Code:

```
// C++ Program to perform mathematical operations on two objects using Binary  
Operator
```

Overloading

```
#include
```

```
<iostream> using
```

```
namespace std;
```

```
class Math
```

```
{
```

```
    int num;
```

```
public:
```

```
    // setter to set value
```

```
void setValue(int val)
```

```
{    num
```

```
= val;
```

```
}
```

```
// overloading + operator to add values in two objects
```

```
    Math operator + (Math
```

```
&obj) {    Math temp;
```

```
temp.num = num + obj.num;
```

```
return (temp);
```

```
}
```

```
// overloading - operator to subtract values in two objects
```

NAME:- Alimehdi Suleman

Roll No: CS24048

Math operator - (Math

&obj){ Math temp;

temp.num = num -

obj.num; return (temp);

}

// overloading * operator to multiply values in two objects

Math operator * (Math

&obj){ Math temp;

temp.num = num *

obj.num; return (temp);

}

// overloading / operator to divide values in two objects

Math operator / (Math

&obj){ Math temp;

temp.num = num /

obj.num; return (temp);

}

// display result value getter

void getValue(){cout << num;

}

};

int main ()

{

// created objects obj1 and obj2 to perform mathematical operations and resObj to

store results Math obj1, obj2, resObj;

NAME:- Alimehdi Suleman
Roll No: CS24048

```
// accepting the
values

obj1.setValue(20);
obj2.setValue(10);

cout << "Obj 1: ";
obj1.getValue();
cout << "\nObj 2: ";
obj2.getValue();

// assign result of obj1 and obj2 to resObj
addition  resObj = obj1 + obj2;  cout <<
"\n\nObj1 + Obj2 : " ;  resObj.getValue(); //
subtraction  resObj = obj1 - obj2;  cout <<
"\n\nObj1 - Obj2 : " ;  resObj.getValue(); //
multiplication  resObj = obj1 * obj2;  cout
<< "\n\nObj1 * Obj2 : " ;  resObj.getValue();

// division  resObj = obj1
/ obj2;  cout << "\n\nObj1
/ Obj2 : " ;
resObj.getValue();

return 0;
}
```

NAME:- Alimehdi Suleman

Roll No: CS24048

Output:-

```
Obj 1: 20
Obj 2: 10

Obj1 + Obj2 : 30
Obj1 - Obj2 : 10
Obj1 * Obj2 : 200
Obj1 / Obj2 : 2
-----
Process exited after 0.04296 seconds with return value 0
Press any key to continue . . . _
```

NAME:- Alimehdi Suleman

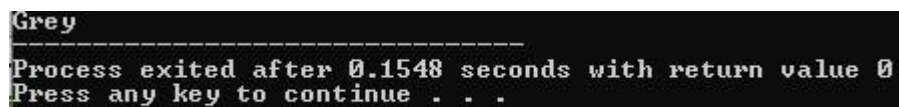
Roll No: CS24048

11.3: Function Overriding

Code:

```
#include <iostream> using
namespace std; class Exam { //
base class declaration.   public:
void colorP(){
cout<<"Black";
}
};
class Atkt: public Exam // inheriting Exam class.
{
public:
void colorP(){
cout<<"Grey";
}
};
int main(void) {
Atkt kt;
kt.colorP();
}
```

Output:-



```
Grey
-----
Process exited after 0.1548 seconds with return value 0
Press any key to continue . . .
```

NAME:- Alimehdi Suleman
Roll No: CS24048

11.4: Pure Virtual Function

Code:

```
#include

<iostream> using
namespace std; //
Abstract class class
Shape
{
    public:    virtual float calculateArea() = 0; // pure
virtual function.
};

class Square : public Shape
{
    float a;
public:
    Square(float l)
    {    a = l;    }    float
calculateArea()
    {    return
a*a;
    }
};

class Circle : public Shape
```

NAME:- Alimehdi Suleman

Roll No: CS24048

```
{  
  
float r;  
  
public:  
  
    Circle(float x) {  
r = x; } float  
calculateArea() {  
return 3.14*r*r;  
}  
};  
  
class Rectangle : public Shape  
{  
float l;  
float b;  
public:  
    Rectangle(float x, float  
y) { l=x; b=y; }  
float calculateArea() {  
return l*b; }  
};  
  
int main()  
{  
    Shape *shape;  
    Square s(3.4);  
    Rectangle r(5,6); Circle c(7.8); shape  
=&s; int a1 =shape->calculateArea();
```

NAME:- Alimehdi Suleman

Roll No: CS24048

shape = &r; int a2 = shape-

>calculateArea(); shape = &c; int a3 =

shape->calculateArea(); cout << "Area of

the square is " <<a1<< endl; cout << "Area

of the rectangle is " <<a2<< endl; cout <<

"Area of the circle is " <<a3<<endl; return

0;

}

Output:-

```
Area of the square is 11
Area of the rectangle is 30
Area of the circle is 191
```

```
-----
Process exited after 0.1719 seconds with return value 0
Press any key to continue . . .
```


NAME:- Alimehdi Suleman

Roll No: CS24048

Practical 12

Program to demonstrate use of This Pointer within a class

Code:

```
#include <iostream>

using namespace std;

class Employee {

public:

    int id; //data member (also instance variable)

    string name; //data member(also instance
variable)    float salary;

    Employee(int id, string name, float salary)

    {    this->id = id;

this->name =

name;    this-

>salary = salary;

    }    void

display()

    {

        cout<<id<<" "<<name<<" "<<salary<<endl;

    }

};

int main(void) {

    Employee e1 =Employee(101, "Pinocchio", 890000); //creating an object of

Employee    Employee e2=Employee(102, "Naruto", 59000); //creating an

object of Employee    e1.display();    e2.display();    return 0;

}
```

NAME:- Alimehdi Suleman

Roll No: CS24048

Output:-

```
101 Pinocchio 890000  
102 Naruto 59000
```

```
-----  
Process exited after 0.04306 seconds with return value 0  
Press any key to continue . . . _
```

Name:Alimehdi Suleman

Roll No: CS24048

Practical 13

Explore the use of pointers within classes, emphasizing dynamic memory allocation.

13.1 Write a c++ program to use the pointers and address of operator and value of operator for pointer variable.

Code:

```
#include <iostream>

using namespace std;

void Pointers()
{
    int var = 20;

    int* ptr;    // declare pointer variable

    ptr = &var;  // data type of ptr and var must be same

    // assign the address of a variable to a pointer

    cout << "Value at ptr = " << ptr << "\n";

    cout << "Value at var = " << var << "\n";

    cout << "Value at *ptr = " << *ptr << "\n";

}

int main()
{
    Pointers();

    return 0;

}
```

Name:Alimehdi Suleman

Roll No: CS24048

Output:

```
Value at ptr = 0x6ffe04
```

```
Value at var = 20
```

```
Value at *ptr = 20
```

```
-----
```

```
Process exited after 7.46 seconds with return value 0
```

```
Press any key to continue . . .
```

Name:Alimehdi Suleman

Roll No: CS24048

13.2 Write a c++ program to use the pointer to pointers

Code:

```
#include<iostream>

using namespace std;

int main ()
{
    int a;
    int * ptr_b;
    int ** ptr_c;
    a = 1;
    ptr_b = &a;
    ptr_c = &ptr_b;           //Get address of ptr_b
    cout << a << "\n";        //print value of a
    cout << *ptr_b << "\n";    //print value where pointer ptr_b points to
    cout << ptr_b << "\n";     //print value of pointer ptr_b
    cout << *ptr_c << "\n";    //print address of ptr_b
    cout << **ptr_c << "\n";  //print value where ptr_c points to
    return 0;
}
```

Output:

```
1
1
0x6ffe44
0x6ffe44
1
-----
Process exited after 0.948 seconds with return value 0
Press any key to continue . . . |
```

Name:Alimehdi Suleman

Roll No: CS24048

13.3 Write a program in C++ to implement the concept of call by value to swap the values of two variables.

Code:

```
#include<iostream>

using namespace std;

void swapping(int c, int d)
{
    int temp;
    temp = c;
    c = d;
    d = temp;
    cout << "In function:\n" << c << " \n " << d << " \n ";
}

int main()
{
    int a,b;
    a=5;
    b=10;
    cout << "Before Sapping :\n  " << a << " \n " << b << " \n ";
    swapping(a,b);
    cout << "After Swapping:\n  " << a << " \n " << b << " \n ";
    return 0;
}
```

Output:

Name:Alimehdi Suleman

Roll No: CS24048

```
Before Sapping :
  5
 10
In function:
10
 5
After Swapping:
 5
10

-----
Process exited after 0.9865 seconds with return value 0
Press any key to continue . . . |
```

13.4 Write a program in C++ to implement the concept of call by reference (using Pointers) to swap the values of two variables.

Code:

```
#include<iostream>

using namespace std;

void swapping(int *ptr_c, int *ptr_d)
{
    int tmp;

    tmp = *ptr_c;
    *ptr_c = *ptr_d;
    *ptr_d = tmp;

    cout << "In function:\n" << *ptr_c << "\n " << *ptr_d << '\n';
}

int main()
{
    int a,b;

    a=5;
    b=10;
```

Name:Alimehdi Suleman

Roll No: CS24048

```
        cout << "Before:\n" << a << " \n" << b << "\n";  
        swapping(&a,&b);  
        cout << "After:\n" << a << "\n " << b << " \n ";  
    }
```

Output:

```
Before:  
5  
10  
In function:  
10  
5  
After:  
10  
5  
  
-----  
Process exited after 0.9054 seconds with return value 0  
Press any key to continue . . . |
```


Name:Alimehdi Suleman

Roll No: CS24048

Practical 14

Develop programs for both text and binary file handling within a class context.

14.1 Write a Program in c++ to create and write in a text file then read from that file

Code:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main(){
    fstream newfile,newfile1;

    newfile.open("testFile.txt",ios::out); // open a file to perform write operation using file
object
    if(newfile.is_open()) //checking whether the file is open
    {
        newfile<<"Hello World from CPP \n"; //inserting text
        newfile.close(); //close the file object
    }

    newfile.open("testFile.txt",ios::in); //open a file to perform read operation using file object
    if (newfile.is_open()){ //checking whether the file is open
        string tp;
        while(getline(newfile, tp)){ //read data from file object and put it into string.
            cout << tp << "\n"; //print the data of the string
        }
        newfile.close(); //close the file object.
    }
}
```

Name:Alimehdi Suleman

Roll No: CS24048

Output:

```
Hello World from CPP
```

```
-----
```

```
Process exited after 0.9322 seconds with return value 0
```

```
Press any key to continue . . . |
```

Name:Alimehdi Suleman

Roll No: CS24048

14.2 Write a program in c++ to write some text to a text File

Code:

```
#include <iostream>    // Include the input/output stream library
#include <fstream>      // Include the file stream library

int main() {
    // Create a new file named "InfoOnCpp.txt"
    std::ofstream outputFile("InfoOnCpp.txt"); // Open/create a file named "test.txt" for writing
    if (outputFile.is_open()) { // Check if the file was successfully opened
        // Write some text into the file
        outputFile << "C++ is a high-level, general-purpose programming language created by
        Danish computer scientist Bjarne Stroustrup. \n"; // Write a line of text to the file
        outputFile << "First released in 1985 as an extension of the C programming language, it has
        since expanded significantly over time. \n"; // Write a line of text to the file
        outputFile << "Modern C++ currently has object-oriented, generic, and functional features,
        in addition to facilities for low-level memory manipulation.\n"; // Write a line of text to the file
        outputFile << "It is almost always implemented in a compiled language.\n"; // Write a line
        of text to the file
        outputFile << "Many vendors provide C++ compilers, including the Free Software
        Foundation, LLVM, Microsoft, Intel, Embarcadero, Oracle, and IBM."; // Write a line of text to
        the file

        // Close the file
        outputFile.close(); // Close the file after writing
        std::cout << "Text has been written to the file." << std::endl; // Display a success message
    } else {
        std::cout << "Failed to create the file." << std::endl; // Display an error message if file
        creation failed
    }
    return 0; // Return 0 to indicate successful execution
}
```

Output:

Name:Alimehdi Suleman

Roll No: CS24048

```
Text has been written to the file.
```

```
-----
```

```
Process exited after 0.9184 seconds with return value 0
```

```
Press any key to continue . . . |
```

Name:Alimehdi Suleman

Roll No: CS24048

14.3 Write a program in c++ to read the same / existing text File

Code:

```
#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main(){

    fstream newfile;


    newfile.open("InfoOnCpp.txt",ios::in); //open a file to perform read operation using file
    object

    if (newfile.is_open()){ //checking whether the file is open

        string tp;

        while(getline(newfile, tp)){ //read data from file object and put it into string.

            cout << tp << "\n"; //print the data of the string

        }

        newfile.close(); //close the file object.

    }

    return 0;

}
```

Output:

Name:Alimehdi Suleman

Roll No: CS24048

14.4 Write a program in c++ to append few lines of text to the same / existing text File

Code:

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ofstream my_file("InfoOnCpp.txt", ios::app); // open a text file for appending
    // if the file doesn't open successfully, print an error message
    if(!my_file) {
        cout << "Failed to open the file for appending." << endl;
        return 1;
    }
    // append multiple lines to the file
    my_file << "Attempted to open file in append mode" << endl;
    my_file << "and type the data" << endl;
    my_file << "It's a Success" << endl;

    my_file.close(); // close the file
    return 0;
}
```

Output:

```
Text has been written to the file.
```

```
C++ is a high-level, general-purpose programming language created by Danish computer scientist Bjarne Stroustrup.
First released in 1985 as an extension of the C programming language, it has since expanded significantly over time.
Modern C++ currently has object-oriented, generic, and functional features, in addition to facilities for low-level memo
ry manipulation.
```

```
It is almost always implemented in a compiled language.
```

```
Many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Embarcadero, Oracle,
and IBM.
```

```
-----
Process exited after 0.9315 seconds with return value 0
```

```
Press any key to continue . . . |
```

Name:Alimehdi Suleman

Roll No: CS24048

14.5 Write a program in c++ to write and read from a binary file.

Code:

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

struct Student
{
    int roll_no;
    string name;
};

int main() {
    ofstream wbf("student.dat", ios::out | ios::binary);

    if (!wbf) {
        cout << "Cannot open file!" << endl;
        return 1;
    }

    Student wstu[3];

    wstu[0].roll_no = 101;
    wstu[0].name = "Ambika";
    wstu[1].roll_no = 102;
    wstu[1].name = "Chandra";
    wstu[2].roll_no = 103;
    wstu[2].name = "Madhu";

    for (int i = 0; i < 3; i++)
        wbf.write((char * ) & wstu[i], sizeof(Student));

    wbf.close();
```

Name:Alimehdi Suleman

Roll No: CS24048

```
    if (!wbf.good())
    {
        cout << "Error occurred at writing time!" << endl;
        return 1;
    }

    ifstream rbf("student.dat", ios::out | ios::binary);
    if (!rbf)
    {
        cout << "Cannot open file!" << endl;
        return 1;
    }

    Student rstu[3];

    for (int i = 0; i < 3; i++)
        rbf.read((char * ) & rstu[i], sizeof(Student));
    rbf.close();
    if (!rbf.good())
    {
        cout << "Error occurred at reading time!" << endl;
        return 1;
    }

    cout << "Student's Details:" << endl;
    for (int i = 0; i < 3; i++)
    {
        cout << "Roll No:  " << wstu[i].roll_no << endl;
        cout << "Name:  " << wstu[i].name << endl;
        cout << endl;
    }

    return 0;
}
```

Output:

Name:Alimehdi Suleman

Roll No: CS24048

Student's Details:

Roll No: 101

Name: Ambika

Roll No: 102

Name: Chandra

Roll No: 103

Name: Madhu