

main.cpp



Run

```
1  #include <iostream>
2  using namespace std;
3  class add {
4  private:
5  int value;
6  public:
7  add(int val)
8  {
9  value=val;
10 }
11 // Overload the + operator
12 add operator+(add& a) {
13 int sm = value + a.value;
14 cout<<"Addition : "<<sm<<endl;
15 }
16 };
17 int main() {
18 add num1(5);
19 add num2(10);
20 add result = num1 + num2;
21 return 0;
22 }
```

Output

Clear

/tmp/sFFQZKLzqj.o

Addition : 15

main.cpp



Run

```
1 #include <iostream>
2 using namespace std;
3 class add {
4 private:
5 int value;
6 public:
7 add(int val)
8 {
9 value=val;
10 }
11 // Overload the + operator
12 friend add operator+(add b, add a);
13 };
14 add operator+(add b, add a) {
15 int sm = b.value + a.value;
16 cout<<"Addition : "<<sm<<endl;
17 }
18 int main() {
19 add num1(5);
20 add num2(10);
21 add result = num1 + num2;
22 return 0;
23 }
```

Output

Clear

/tmp/sFFQZKLzqj.o

Addition : 15

|

main.cpp



Run

```
1 #include <iostream>
2 using namespace std;
3 // Base class
4 class Shape {
5 public:
6 virtual void draw() {
7     cout << "Drawing a shape." <<endl;
8 }
9 };
10 // Derived class 1
11 class Circle : public Shape {
12 public:
13 void draw() override{
14     cout << "Drawing a circle." <<endl;
15 }
16 };
17 // Derived class 2
18 class Square : public Shape {
19 public:
20 void draw() override{
21     cout << "Drawing a square." <<endl;
22 }
23 };
24 int main() {
25     // Create objects of base and derived classes
26     Shape shape;
27     Circle circle;
```

main.cpp



Run

```
12 public:
13 void draw() override{
14     cout << "Drawing a circle." <<endl;
15 }
16 };
17 // Derived class 2
18 class Square : public Shape {
19 public:
20 void draw() override{
21     cout << "Drawing a square." <<endl;
22 }
23 };
24 int main() {
25     // Create objects of base and derived classes
26     Shape shape;
27     Circle circle;
28     Square square;
29     // Create pointers to base class objects
30     Shape* shapePtr1 = &shape;
31     Shape* shapePtr2 = &circle;
32     Shape* shapePtr3 = &square;
33     // Call the draw() function through base class pointers
34     shapePtr1->draw(); // Output: Drawing a shape.
35     shapePtr2->draw(); // Output: Drawing a circle.
36     shapePtr3->draw(); // Output: Drawing a square.
37     return 0;
38 }
```

Output

Clear

```
/tmp/sFFQZKLzqj.o
```

```
Drawing a shape.
```

```
Drawing a circle.
```

```
Drawing a square.
```

main.cpp



Run

```
1  #include <iostream>
2  using namespace std;
3  class Negation {
4  public:
5      int x, y;
6      Negation(int a, int b)
7      {
8          x = a;
9          cout<<"x : "<<x<<endl;
10 }
11 //overload - operator
12 void operator-()
13 {
14     x=++x;
15     cout<<endl<<"After performing unary operator overlaoding"<<endl;
16     cout<<"x : "<<x<<endl;
17 }
18 };
19 int main()
20 {
21     Negation d1(8, 9);
22     -d1;
23     return 0;
24 }
```


Output

Clear

/tmp/sFFQZKLzqj.o

x : 8

After performing unary operator overlaoding

x : 9

|

main.cpp



Run

```
1 #include <iostream>
2 using namespace std;
3 class Negation {
4 public:
5     int x, y;
6     Negation(int a, int b)
7     {
8         x = a;
9         cout<<"x : "<<x<<endl;
10    }
11    //overload - operator
12    friend void operator-(Negation &N);
13    };
14    void operator-(Negation &N)
15    {
16        N.x=++N.x;
17        cout<<endl<<"After performing unary operator overlaoding"<<endl;
18        cout<<"x : "<<N.x<<endl;
19    }
20    int main()
21    {
22        Negation d1(8, 9);
23        -d1;
24        return 0;
25    }
```

Output

Clear

/tmp/sFFQZKLzqj.o

x : 8

After performing unary operator overlaoding

x : 9

|

main.cpp



Run

```
1 #include<iostream>
2 using namespace std;
3 class addition{
4 public:
5 // Function to add two integers
6 void add(int a, int b) {
7 cout<<"Sum of int values : "<<a+b<<endl;
8 }
9 // Function to add two doubles
10 void add(double a, double b) {
11 cout<<"Sum of double values : "<<a+b<<endl;
12 }
13 };
14 int main() {
15 addition dd;
16 // Calls int add(int a, int b)
17 dd.add(5, 3);
18 // Calls double add(double a, double b)
19 dd.add(2.5, 3.7);
20 return 0;
21 }
```

Output

Clear

/tmp/sFFQZKLzqj.o

Sum of int values : 8

Sum of double values : 6.2

main.cpp



Run

```
1 // Online C++ compiler to run C++ program online
2 #include <iostream>
3 using namespace std;
4 int main() {
5     // Write C++ code here
6     int n,d,r;
7     cout<<endl<<"enter numerator : ";
8     cin>>n;
9     cout<<endl<<"enter denominator : ";
10    cin>>d;
11    try
12    {
13        if(d==0)
14        {
15            throw d;
16        }
17        r=n/d;
18        cout<<"result : "<<r<<endl;
19    }
20    catch(int exception)
21    {
22        cout<<"Exception : Division by zero"<<endl;
23    }
24 }
```

Output

Clear

/tmp/sFFQZKLzqj.o

enter numerator : 6

enter denominator : 8

result : 0

main.cpp



Run

```
1  #include<iostream>
2  using namespace std;
3  template <class T1>
4  void fun(T1 val)
5  {
6  cout<<"Result : "<<val<<endl;
7  }
8  int main()
9  {
10 fun('C'); //for character
11 fun("C++"); //for string
12 fun(56); //for integer
13 fun(89.9); //for float
14 }
```


Output

Clear

/tmp/sFFQZKLzqj.o

enter numerator : 6

enter denominator : 8

result : 0

main.cpp



Run

```
1  #include<iostream>
2  using namespace std;
3  template <class T1>
4  class temp{
5  public:
6  T1 value;
7  temp(T1 val)
8  {
9  value=val;
10 cout<<"Result : "<<val<<endl;
11 }
12 };
13 int main()
14 {
15 temp <char> t('C'); // for character
16 temp <string> u("C++"); // for string
17 temp <int> v(58); //for integer
18 temp <float> w(98.9); //for float
19 }
```

Output

Clear

/tmp/sFFQZKLzqj.o

Result : C

Result : C++

Result : 58

Result : 98.9

main.cpp



Run

```
1  #include<iostream>
2  using namespace std;
3  // Function to add two integers
4  int add(int a, int b) {
5      return a + b;
6  }
7  // Function to add two doubles
8  double add(double a, double b) {
9      return a + b;
10 }
11 int main() {
12     // Calls int add(int a, int b)
13     int addition_int_values = add(5, 3);
14     // Calls double add(double a, double b)
15     double addition_double_values = add(2.5, 3.7);
16     cout << "Addition of int values: " << addition_int_values << endl;
17     cout << "Addition of double values: " << addition_double_values << endl;
18     return 0;
19 }
```

Output

Clear

/tmp/sFFQZKLzqj.o

Addition of int values: 8

Addition of double values: 6.2

|

main.cpp

Run

```
1  #include<iostream>
2  using namespace std;
3  class temp{
4  public:
5      template <class T1>
6      char fun(T1 val)
7      {
8          cout<<"Result : "<<val<<endl;
9      }
10 };
11 int main()
12 {
13     temp t;
14     t.fun("C++"); //for string
15     t.fun('C'); // for character
16     t.fun(56); // for integer
17     t.fun(9.8); // for float
18 }
```

v

Output

Clear

/tmp/sFFQZKLzqj.o

Result : C++

Result : C

Result : 56

Result : 9.8