

Mcs21

Exception Handling

```
#include <iostream>
#include <stdexcept>
using namespace std;
int main()
{try {
    int numerator = 10;
    int denominator = 0;
    int res; if (denominator == 0) {
        throw runtime_error(
            "Division by zero not allowed!");    }
    res = numerator / denominator;
    cout << "Result after division: " << res <<
endl; }
    catch (const exception& e) {
        // print the exception
        cout << "Exception " << e.what() << endl; }
    return 0; }
```

Function Overloading

```
#include <iostream>
using namespace std;
class Cal { public:
static int add(int a,int b){
    return a + b; }
static int add(int a, int b, int c)
{    return a + b + c; } };
int main(void) {
    Cal C;
    cout<<C.add(10, 20)<<endl;
    cout<<C.add(12, 20, 23);
    return 0; }
```

Output:

30
55

Operator Overloading

```
#include <iostream>
using namespace std;
class Test {
    private:
        int num;
    public:
        Test(): num(8){}
        void operator ++() {
            num = num+2; }
        void Print() {
            cout<<"The Count is: "<<num; } };
int main()
{    Test tt;
    ++tt; // calling of a function "void operator ++()"

    tt.Print();
    return 0; }
Output:
The Count is: 10
```

Constructor

```
#include <bits/stdc++.h>
using namespace std;
class Employee {
    public:
        int age;
        Employee() {
            age = 50; };
int main() {
    Employee e1;
    cout << e1.age;
    return 0;
}
```

Destructor

```
#include <iostream>
using namespace std;
class Employee
{ public:
    Employee()
    { cout<<"Constructor Invoked"<<endl; }

    ~Employee()
    { cout<<"Destructor Invoked"<<endl; }
};
int main(void)
{ Employee e1; //creating an object of Employee
  Employee e2; //creating an object of Employee
  return 0; }
```

Inline function

```
#include <iostream>
// Inline function to calculate the square of a
number
inline int square(int x) {
    return x * x;
}
int main() {
    int num = 5;

    // Call the inline function
    int result = square(num);

    // Output the result
    std::cout << "The square of " << num << " is: " << result << std::endl;

    return 0;
}
```

C++ Code Example 2 num x

```
#include <iostream>
int main() {
    // Define the two numbers to multiply
    int num1 = 5;
    int num2 = 7;

    // Calculate the product
    int product = num1 * num2;

    // Output the result
    std::cout << "The product of " << num1 << "
and " << num2 << " is: " << product <<
std::endl;

    return 0;
}
```

Friend function

```
#include <iostream>
class MyClass;
void showValue(const MyClass&);
class MyClass {
private:
    int value;
public:
    MyClass(int v) : value(v) {}
    // Friend declaration
    friend void showValue(const MyClass&);
};void showValue(const MyClass& obj) {
    std::cout << "The value of MyClass is: " <<
obj.value << std::endl;
}
int main() {
    MyClass obj(42);
    // Call the friend function
    showValue(obj);
    return 0;
}
```