1. Class and Object Basics

```
#include <iostream>
using namespace std;
class Student {
public:
    string name;
    int roll_no;
    void getDetails() {
        cout << "Enter name: ";
        cin >> name;
        cout << "Enter roll number: ";
        cin >> roll_no;
    }
    void display() {
        cout << "Name: " << name << endl;
        cout << "Roll No: " << roll_no << endl;
    }
};
```

2. Constructor Initialization

```
#include <iostream>
using namespace std;
class Book {
public:
    string title;
    float price;
    Book(string t, float p) {
        title = t;
        price = p;
    }
    void display() {
        cout << "Title: " << title << endl;
        cout << "Price: " << price << endl;
    }
};
```

3. Single Inheritance

```
#include <iostream>
using namespace std;
class Person {
public:
    string name;
};
class Employee : public Person {
public:
    int salary;
    void display() {
        cout << "Name: " << name << endl;
        cout << "Salary: " << salary << endl;
    }
};
```

4. Multilevel Inheritance

```
#include <iostream>
using namespace std;
class Vehicle {
public:
    string brand;
};
class Car : public Vehicle {
public:
    string model;
};
class ElectricCar : public Car {
public:
    int battery;
    void show() {
         cout << "Brand: " << brand << ", Model: " << model << ", Battery: " << battery
<< "%" << endl;
    }
};
```

5. Method Overloading

```
#include <iostream>
using namespace std;
class Calculator {
public:
    int add(int a, int b) { return a + b; }
    int add(int a, int b, int c) { return a + b + c; }
    float add(float a, float b) { return a + b; }
};
```

6. Method Overriding & Polymorphism

```
#include <iostream>
using namespace std;
class Shape {
public:
    virtual void draw() {
        cout << "Drawing Shape" << endl;
    }
};
class Circle : public Shape {
public:
    void draw() override {
        cout << "Drawing Circle" << endl;
    }
};
class Rectangle : public Shape {
public:
    void draw() override {
        cout << "Drawing Rectangle" << endl;
    }
};
```

7. Operator Overloading (+)

```
#include <iostream>
using namespace std;
class Complex {
public:
    int real, imag;
    Complex(int r = 0, int i = 0) : real(r), imag(i) {}
    Complex operator+(const Complex &obj) {
        return Complex(real + obj.real, imag + obj.imag);
    }
};
```