

AAGASC – Department of Computer Science
Subject: Object Oriented Programming Lab using C++

1. Program to Read and Print Number Input from the User

```
#include <iostream.h>
#include <conio.h>

void main() {
    clrscr();
    int num;
    cout << "Enter a number: ";
    cin >> num;
    cout << "You entered: " << num;
    getch();
}
```

Output:

Enter a number: 25

You entered: 25

2. Simple Program using Class and Objects

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class Student {
```

```
    int rollno;
```

```
    char name[20];
```

```
    int m1, m2, m3;
```

```
    int total;
```

```
    float average;
```

```
public:
```

```
    void getData() {
```

```
        cout << "Enter Roll Number: ";
```

```
        cin >> rollno;
```

```
        cout << "Enter Name: ";
```

```
        cin >> name;
```

```
        cout << "Enter 3 Subject Marks: ";
```

```
        cin >> m1 >> m2 >> m3;
```

```
    }
```

```
    void calculate() {
```

```
        total = m1 + m2 + m3;
```

```
        average = total / 3.0;
```

```
    }
```

```
    void display() {
```

```
        cout << "\n--- Student Information ---\n";
```

```
        cout << "Roll No  : " << rollno << "\n";
```

```
        cout << "Name    : " << name << "\n";
```

```
        cout << "Marks   : " << m1 << ", " << m2 << ", " << m3 << "\n";
```

```
        cout << "Total    : " << total << "\n";
```

```
        cout << "Average  : " << average << "\n";
```

```
    }
```

```
};
```

```
void main() {  
    clrscr();  
    Student s;  
    s.getData();  
    s.calculate();  
    s.display();  
    getch();  
}
```

Input and Output:

Enter Roll Number: 101

Enter Name: Anitha

Enter 3 Subject Marks: 78 85 90

--- Student Information ---

Roll No : 101

Name : Anitha

Marks : 78, 85, 90

Total : 253

Average : 84.3333

3. Constructor and Destructor

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class Student {
```

```
    int rollno;
```

```
    char name[20];
```

```
    int m1, m2, m3;
```

```
    int total;
```

```
    float average;
```

```
public:
```

```
    // Constructor
```

```
    Student(int r, char n[], int a, int b, int c) {
```

```
        rollno = r;
```

```
        strcpy(name, n); // copy string (Turbo C++ requires <string.h>)
```

```
        m1 = a;
```

```
        m2 = b;
```

```
        m3 = c;
```

```
        total = m1 + m2 + m3;
```

```
        average = total / 3.0;
```

```
    }
```

```
    void display() {
```

```
        cout << "\n--- Student Information ---\n";
```

```
        cout << "Roll No   : " << rollno << "\n";
```

```
        cout << "Name     : " << name << "\n";
```

```
        cout << "Marks    : " << m1 << ", " << m2 << ", " << m3 << "\n";
```

```
        cout << "Total    : " << total << "\n";
```

```
        cout << "Average  : " << average << "\n";
```

```
    }
```

```
};
```

```
void main() {
```

```
    clrscr();
```

```
// Creating student object with constructor
Student s1(101, "Arun", 80, 75, 90);
Student s2(102, "Kavya", 60, 85, 70);

s1.display();
s2.display();

getch();
}
```

Output:

--- Student Information ---

Roll No : 101

Name : Arun

Marks : 80, 75, 90

Total : 245

Average : 81.6667

--- Student Information ---

Roll No : 102

Name : Kavya

Marks : 60, 85, 70

Total : 215

Average : 71.6667

4. Operator Overloading (+ for Complex Numbers)

```
#include <iostream.h>
#include <conio.h>
class Complex {
    int real, imag;
public:
    void getData() {
        cout << "Enter real and imaginary part: ";
        cin >> real >> imag;
    }
    void display() {
        cout << real << " + " << imag << "i";
    }
    Complex operator+(Complex c) {
        Complex temp;
        temp.real = real + c.real;
        temp.imag = imag + c.imag;
        return temp;
    }
};

void main() {
    clrscr();
    Complex c1, c2, c3;
    cout << "Enter first complex number:\n";    c1.getData();
    cout << "Enter second complex number:\n";   c2.getData();
    c3 = c1 + c2;
    cout << "Sum = ";
    c3.display();
    getch();
}
```

Input & Output:

Enter first complex number:

Enter real and imaginary part: 3 4

Enter second complex number:

Enter real and imaginary part: 5 6

5. Function Overloading

```
#include <iostream.h>
#include <conio.h>

class Print {
public:
    void show(int a) {
        cout << "Integer: " << a << "\n";
    }
    void show(float b) {
        cout << "Float: " << b << "\n";
    }
    void show(char c) {
        cout << "Character: " << c << "\n";
    }
};

void main() {
    clrscr();
    Print p;
    p.show(10);
    p.show(3.14f);
    p.show('A');
    getch();
}
```

Output:

Integer: 10

Float: 3.14

Character: A

6. Multiple Inheritance (Employee Information)

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class Personal {  
protected:  
    char name[20];  
    int age;  
public:  
    void getPersonal() {  
        cout << "Enter Name: ";  
        cin >> name;  
        cout << "Enter Age: ";  
        cin >> age;  
    }  
};
```

```
class Official {  
protected:  
    char company[20];  
    int experience;  
public:  
    void getOfficial() {  
        cout << "Enter Company Name: ";  
        cin >> company;  
        cout << "Enter Years of Experience: ";  
        cin >> experience;  
    }  
};
```

```
class Employee : public Personal, public Official {  
    int empID;  
public:  
    void getData() {  
        cout << "Enter Employee ID: ";  
        cin >> empID;  
    }  
};
```

```

        getPersonal();
        getOfficial();
    }
    void display() {
        cout << "\n--- Employee Information ---\n";
        cout << "Employee ID : " << empID << "\n";
        cout << "Name      : " << name << "\n";
        cout << "Age       : " << age << "\n";
        cout << "Company    : " << company << "\n";
        cout << "Experience  : " << experience << " years\n";
    }
};

void main() {
    clrscr();
    Employee e;
    e.getData();
    e.display();
    getch();
}

```

Input & Output:

```

Enter Employee ID: 101
Enter Name: Karthik
Enter Age: 28
Enter Company Name: Infosys
Enter Years of Experience: 5

```

```

--- Employee Information ---
Employee ID : 101
Name      : Karthik
Age       : 28
Company    : Infosys
Experience  : 5 years

```

7. Multilevel Inheritance

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class Person {  
protected:  
    char name[20];  
public:  
    void getName() {  
        cout << "Enter name: ";  
        cin >> name;  
    }  
};
```

```
class Student : public Person {  
protected:  
    int roll;  
public:  
    void getRoll() {  
        cout << "Enter roll number: ";  
        cin >> roll;  
    }  
};
```

```
class Result : public Student {  
    int marks;  
public:  
    void getMarks() {  
        cout << "Enter marks: ";  
        cin >> marks;  
    }  
    void display() {  
        cout << "Name: " << name << "\n";  
        cout << "Roll Number: " << roll << "\n";  
        cout << "Marks: " << marks << "\n";  
    }  
};
```

```
};
```

```
void main() {  
    clrscr();  
    Result r;  
    r.getName();  
    r.getRoll();  
    r.getMarks();  
    r.display();  
    getch();  
}
```

Input & Output:

Enter name: Priya

Enter roll number: 105

Enter marks: 89

Name: Priya

Roll Number: 105

Marks: 89