

E-Content Module

Title:

Understanding Class and Object in C++

Introduction

In Object-Oriented Programming (OOP), the two most important building blocks are **class** and **object**. They help us organise code in a logical and structured way. Instead of writing scattered functions and variables, we group related data and behaviour together.

Think of real life. A *car* has properties like colour, model, and speed, and it performs actions like start, stop, and accelerate. Similarly, in C++, we can create a structure that represents real-world entities. That structure is called a **class**, and the real usable item created from it is called an **object**.

Understanding classes and objects is the first and most important step in mastering C++ OOP concepts.

Learning Objectives

After completing this module, learners will be able to:

- Define a class in C++
 - Create objects from a class
 - Understand data members and member functions
 - Differentiate between class and object
 - Write simple C++ programs using classes and objects
-

Content Presentation

1. What is a Class?

A **class** is a blueprint or template used to create objects.

It contains:

- **Data Members** → Variables that store data
- **Member Functions** → Functions that define behaviour

Syntax of Class in C++

```
class ClassName {  
    private:  
        // data members  
    public:  
        // member functions  
};
```

2. What is an Object?

An **object** is an instance of a class.

If class is a blueprint of a house, then an object is the actual house built from that blueprint.

3. Example 1: Student Class

Let us create a simple class to understand clearly.

```
#include<iostream>  
using namespace std;  
  
class Student {  
private:  
    string name;  
    int roll;  
  
public:  
    void setData(string n, int r) {  
        name = n;  
        roll = r;  
    }  
  
    void display() {  
        cout << "Name: " << name << endl;  
        cout << "Roll: " << roll << endl;  
    }  
};  
  
int main() {  
    Student s1;          // Object creation  
  
    s1.setData("Rahul", 101);  
    s1.display();  
}
```

```
    return 0;  
}
```

Explanation:

- `class Student` → Defines a class
 - `name` and `roll` → Data members
 - `setData()` and `display()` → Member functions
 - `Student s1;` → Object creation
 - `s1.setData()` → Accessing class functions using object
-

4. Access Specifiers

C++ provides three access specifiers:

Specifier	Meaning
<code>private</code>	Accessible only inside the class
<code>public</code>	Accessible outside class
<code>protected</code>	Used in inheritance

In most cases:

- Data members → `private`
- Functions → `public`

This ensures **data security (Encapsulation)**.

5. Example 2: Bank Account

```
#include<iostream>  
using namespace std;  
  
class BankAccount {  
private:  
    string name;  
    double balance;  
  
public:
```

```

void createAccount(string n, double b) {
    name = n;
    balance = b;
}

void deposit(double amount) {
    balance += amount;
}

void showBalance() {
    cout << "Account Holder: " << name << endl;
    cout << "Current Balance: " << balance << endl;
}
};

int main() {
    BankAccount acc1;

    acc1.createAccount("Anita", 5000);
    acc1.deposit(2000);
    acc1.showBalance();

    return 0;
}

```

What Happens Here?

1. Class defines structure.
2. Object acc1 is created.
3. Data is assigned using a member function.
4. Balance is modified using an object.

This shows how **data and functions work together inside a class**.

Class vs Object (Important Difference)

Class	Object
Blueprint	Instance
Logical entity	Physical entity
No memory allocated	Memory allocated
Example: Car	Example: MyCar

Activities

1. Create a class named `Book` with:
 - Data members: `title`, `price`
 - Member functions: `setBook()`, `displayBook()`
 2. Modify the `Student` program to create two objects.
 3. Create a class `Rectangle` and calculate the area.
-

Assessment

Short Answer Questions

1. What is a class in C++?
2. Define an object with an example.
3. What is the difference between a class and an object?
4. Why are data members usually kept private?

Practical Question

Write a C++ program to create a class `Employee` with:

- `empName`
 - `empId`
 - `salary`
- Include functions to set and display data.
-

Summary

In this module, we learned that:

- A **class** is a blueprint that defines data and behaviour.
- An **object** is an instance of a class.
- Data members store information.
- Member functions define actions.

- Access specifiers help in data security.

Classes and objects are the foundation of Object-Oriented Programming in C++. Once this concept is clear, advanced topics like constructors, inheritance, and polymorphism become much easier.

References

1. Bjarne Stroustrup – *The C++ Programming Language*
2. E. Balagurusamy – *Object Oriented Programming with C++*
3. C++ Official Documentation – <https://cplusplus.com>