Tutorial 54 - Polymorphism in C++

Definition:

- **Polymorphism**: The ability of a single function or operator to take multiple forms. Derived from "Poly" (many) and "Morphism" (forms).
- Two types:
  a. **Compile-Time Polymorphism (Early Binding)**
  b. **Run-Time Polymorphism (Late Binding)**

---

1. Compile-Time Polymorphism:

- **Definition**: The function to be executed is determined during compilation.
- **Types**:
  a. **Function Overloading**:
    - Multiple functions with the same name but different parameter lists.
    - Example:

```
class Calculator {
public:
    int add(int a, int b) { return a + b; }
    float add(float a, float b) { return a + b; }
};

```

  b. **Operator Overloading**:
    - Custom functionality for operators.
    - Example:

```
class String {
    string s;
public:
    String(string str) : s(str) {}
    String operator+(String obj) {
        return String(s + obj.s);
    }
};

```

---

2. Run-Time Polymorphism:

- **Definition**: The function to be executed is determined at runtime.
- **Key Feature**: Achieved using **virtual functions**.
- **Virtual Function**:
  - Declared in the base class using the `virtual` keyword.
  - Overridden in the derived class.
  - Enables late binding (resolved during runtime).
  - Example:

```cpp
class Base {
public:
    virtual void display() {
        cout << "Base class display" << endl;
    }
};

class Derived : public Base {
public:
    void display() override {
        cout << "Derived class display" << endl;
    }
};

int main() {
    Base* basePtr;
    Derived derivedObj;
    basePtr = &derivedObj;
    basePtr->display(); // Outputs: "Derived class display"
    return 0;
}

```

Key Points:

1. **Compile-Time Polymorphism**:
   - Faster as resolved during compilation.
   - Examples: Function Overloading, Operator Overloading.
2. **Run-Time Polymorphism**:
   - Flexible but slower due to runtime decision-making.
   - Achieved using virtual functions.
3. **Virtual Functions**:
   - Must be declared in the base class.
   - Allow dynamic method dispatch.

Short Notes for Notebook:

Polymorphism in C++:

- **Definition**: One name, multiple forms.
- **Types**:
  a. **Compile-Time Polymorphism** (Early Binding):
     - Resolved during compilation.
     - Examples:
       - **Function Overloading**: Same function name, different parameter lists.
       - **Operator Overloading**: Custom functionality for operators.
  b. **Run-Time Polymorphism** (Late Binding):
     - Resolved during runtime.
     - Achieved using **virtual functions**.

Virtual Function:

- Declared using `virtual` keyword in the base class.
- Overridden in the derived class.
- Enables dynamic binding for function calls.