Tutorial 36 - Inheritance in C++

Overview

1. **Reusability**:
   - A key feature of Object-Oriented Programming (OOPs).
   - Allows reuse of existing classes and addition of new features.
   - Saves time, effort, and cost.
   - Tested and debugged classes ensure reliability.
2. **Definition**:
   - Inheritance is the mechanism of deriving a new class (derived class) from an existing class (base class).
   - The derived class inherits the properties and behaviors of the base class.

Types of Inheritance

1. Single Inheritance

- **Description**:
  - A single derived class inherits from one base class.
- **Example**:
  - `Programmer` class inherits from `Employee` class.

```
class Employee {
    // Base class properties and methods
};
class Programmer : public Employee {
    // Programmer inherits Employee properties
};

```

---

2. Multiple Inheritance

- **Description**:
  - A single derived class inherits from multiple base classes.
- **Example**:
  - `Programmer` class inherits from `Employee` and `Assistant` classes.

```
class Employee {
    // Employee properties
};
class Assistant {
    // Assistant properties
};
class Programmer : public Employee, public Assistant {
    // Programmer inherits from both Employee and Assistant
};

```

---

3. Hierarchical Inheritance

- **Description**:
  - Multiple derived classes inherit from a single base class.
- **Example**:
  - `Programmer` and `Manager` inherit from `Employee`.

```
class Employee {
    // Employee properties
};
class Programmer : public Employee {
    // Programmer inherits Employee properties
};
class Manager : public Employee {
    // Manager inherits Employee properties
};

```

---

4. Multilevel Inheritance

- **Description**:
  - A derived class inherits from another derived class.
- **Example**:
  - `Mammal` inherits from `Animal`, and `Cow` inherits from `Mammal`.

```
class Animal {
    // Animal properties
};
class Mammal : public Animal {
    // Mammal inherits Animal properties
};
class Cow : public Mammal {
    // Cow inherits Mammal properties
};

```

---

5. Hybrid Inheritance

- **Description**:
  - A combination of multiple and multilevel inheritance.
- **Example**:
  - `Mammal` and `Bird` inherit from `Animal`. `Bat` inherits from `Mammal` and `Bird`.

```
class Animal {
    // Animal properties
};
class Mammal : public Animal {
    // Mammal inherits Animal
};
class Bird : public Animal {
    // Bird inherits Animal
};
class Bat : public Mammal, public Bird {
    // Bat inherits Mammal and Bird properties
```

```
};

```

Short Notes for Notebook

Inheritance in C++

1. **Definition**:
   - Mechanism of creating new classes (derived) from existing classes (base).
   - Promotes reusability and extensibility.
2. **Types**:
   - **Single**: One derived class inherits one base class.
   - **Multiple**: One derived class inherits multiple base classes.
   - **Hierarchical**: Multiple derived classes inherit one base class.
   - **Multilevel**: A derived class inherits another derived class.
   - **Hybrid**: Combination of multiple and multilevel inheritance.
3. **Example**:

```
class Base {
    // Base class code
};
class Derived : public Base {
    // Derived class inherits Base
};

```

4. **Key Benefits**:
   - Saves time and cost.
   - Ensures reusability of tested and debugged code.
   - Facilitates scalability.