Tutorial 40 - Multilevel Inheritance in C++

Definition:

- **Multilevel inheritance** occurs when a derived class is inherited from another derived class, forming a chain of inheritance.
- Example:
  - Class `Animal` → Class `Mammal` (inherits `Animal`) → Class `Cow` (inherits `Mammal`).
  - `Mammal` inherits functionalities from `Animal`, and `Cow` inherits functionalities from `Mammal`.

Code Explanation

Code Snippet 1: `Student` Class

```
class Student {
protected:
    int roll_number;
public:
    void set_roll_number(int);
    void get_roll_number(void);
};
void Student::set_roll_number(int r) {
    roll_number = r;
}
void Student::get_roll_number() {
    cout << "The roll number is " << roll_number << endl;
}

```

Key Points:

1. Contains:
   - **Protected** member: `roll_number`.
   - Public functions: `set_roll_number()` and `get_roll_number()`.
2. **Functions**:
   - `set_roll_number`: Sets the value of `roll_number`.
   - `get_roll_number`: Prints the value of `roll_number`.

Code Snippet 2: `Exam` Class

```
class Exam : public Student {
protected:
    float maths, physics;
public:
    void set_marks(float, float);
    void get_marks(void);
};
void Exam::set_marks(float m1, float m2) {
    maths = m1;
    physics = m2;
}
void Exam::get_marks() {
```

```
    cout << "Marks in Maths: " << maths << endl;
    cout << "Marks in Physics: " << physics << endl;
}

```

Key Points:

1. Inherits `Student` publicly.
2. Contains:
   - **Protected** members: `maths` and `physics`.
   - Public functions: `set_marks()` and `get_marks()`.
3. **Functions**:
   - `set_marks`: Sets values of `maths` and `physics`.
   - `get_marks`: Prints values of `maths` and `physics`.

Code Snippet 3: `Result` Class

```
class Result : public Exam {
    float percentage;
public:
    void display_results() {
        get_roll_number();
        get_marks();
        cout << "Your result is: " << (maths + physics) / 2 << "%" << endl;
    }
};

```

Key Points:

1. Inherits `Exam` publicly.
2. Contains:
   - **Private** member: `percentage`.
   - Public function: `display_results()`.
3. **Function**:
   - `display_results`:
     - Calls `get_roll_number()` and `get_marks()`.
     - Calculates percentage as `(maths + physics) / 2`.
     - Prints the result.

Code Snippet 4: Main Program

```
int main() {
    Result harry;
    harry.set_roll_number(420);
    harry.set_marks(94.0, 90.0);
    harry.display_results();
    return 0;
}

```

Key Points:

1. Creates a `Result` object `harry`.

2. Calls:
   - `set_roll_number(420)` → Sets roll number.
   - `set_marks(94.0, 90.0)` → Sets marks in `maths` and `physics`.
   - `display_results()` → Displays roll number, marks, and percentage.

Short Notes for Notebook

Multilevel Inheritance:

1. A derived class is inherited from another derived class.
2. Enables a chain of inheritance (e.g., `Student` → `Exam` → `Result`).

Code Example:

1. `Student` **Class**:
   - **Protected**: `roll_number`.
   - Functions: `set_roll_number()` (sets roll number), `get_roll_number()` (prints roll number).
2. `Exam` **Class**:
   - **Protected**: `maths`, `physics`.
   - Functions:
     - `set_marks()` → Sets marks.
     - `get_marks()` → Prints marks.
3. `Result` **Class**:
   - **Private**: `percentage`.
   - Function:
     - `display_results()`:
       - Calls `get_roll_number()` and `get_marks()`.
       - Calculates percentage as `(maths + physics) / 2`.

Main Function:

1. Create `Result` object.
2. Call:
   - `set_roll_number()` → Assign roll number.
   - `set_marks()` → Assign marks.
   - `display_results()` → Display roll number, marks, and percentage.