Tutorial 22 - OOPs Recap & Nesting of Member Functions in C++

Object-Oriented Programming Recap

1. **Origin**: C++ was originally called "C with Classes" by Bjarne Stroustrup.
2. **Classes vs. Structures**:
   - Structures in C:
     - Members are public.
     - No methods.
   - Classes in C++:
     - Extend structures with additional features.
     - Support methods and properties.
     - Allow public and private access control.
3. **Object Declaration with Class**:
   - Objects can be declared alongside the class definition:

```
class Employee {
    // Class definition
} harry, rohan, lovish;

```

Nesting of Member Functions

- **Definition**: Calling one member function inside another member function of the same class.
- **Example Class**:

```
class binary {
private:
    string s;
    void chk_bin(void); // Private member function
public:
    void read(void);
    void ones_compliment(void);
    void display(void);
};

```

Function Definitions

1. **Read Function**:
   - Takes input from the user.

```
void binary::read(void) {
    cout << "Enter a binary number" << endl;
    cin >> s;
}

```

2. **Check Binary Function**:

- Validates that the input string is a binary number.

```
void binary::chk_bin(void) {
    for (int i = 0; i < s.length(); i++) {
        if (s.at(i) != '0' && s.at(i) != '1') {
            cout << "Incorrect binary format" << endl;
            exit(0);
        }
    }
}

```

3. **One's Compliment Function**:
   - Calls `chk_bin` (nested function) to validate input.
   - Converts `0` to `1` and `1` to `0`.

```
void binary::ones_compliment(void) {
    chk_bin();
    for (int i = 0; i < s.length(); i++) {
        if (s.at(i) == '0') {
            s.at(i) = '1';
        } else {
            s.at(i) = '0';
        }
    }
}

```

4. **Display Function**:
   - Prints the binary string.

```
void binary::display(void) {
    cout << "Displaying your binary number" << endl;
    for (int i = 0; i < s.length(); i++) {
        cout << s.at(i);
    }
    cout << endl;
}

```

Main Function

- Demonstrates the use of the class and nested functions.

```
int main() {
    binary b;
    b.read();             // Takes binary input
    b.display();          // Displays original binary number
    b.ones_compliment();  // Converts to one's complement
    b.display();          // Displays one's complement
    return 0;
}

```

Key Points

1. **Private Members**:
   - `chk_bin` is private and can only be accessed by member functions of the class.
2. **Nesting**:
   - The `chk_bin` function is called inside the `ones_compliment` function.
3. **Output**:
   - Validates and transforms binary input:

```
Enter a binary number
1010
Displaying your binary number
1010
Displaying your binary number
0101

```

Short Notes

1. **Nesting of Member Functions**:
   - One member function is called inside another within the same class.
2. **Class Example**:

```
class binary {
private:
    string s;
    void chk_bin();  // Validates binary input
public:
    void read();     // Reads binary input
    void ones_compliment();  // Calculates one's complement
    void display();  // Displays binary number
};

```

3. **Function Highlights**:
   - **read**: Takes binary input.
   - **chk_bin**: Ensures input contains only `0` and `1`.
   - **ones_compliment**: Flips `0` to `1` and vice versa (calls `chk_bin`).
   - **display**: Prints the binary number.
4. **Code Flow**:
   - Read → Validate (chk_bin) → Transform (ones_compliment) → Display.
5. **Important Syntax**:
   - Private functions can only be accessed within the class.
   - Nesting is useful for modular and reusable code.