

Tutorial 25 - Array of Objects & Passing Objects as Function Arguments in C++

1. Array of Objects

1. Definition:

- An array of objects consists of class objects as its elements.
- Declared like other data-type arrays but contains objects.

2. Example:

```
1 class Employee {
2     int id;
3     int salary;
4 public:
5     void setId(void) {
6         salary = 122;
7         cout << "Enter the id of employee" << endl;
8         cin >> id;
9     }
10    void getId(void) {
11        cout << "The id of this employee is " << id << endl;
12    }
13 };
14
```

3. Usage in main:

```
1 int main() {
2     Employee fb[4]; // Array of 4 Employee objects
3     for (int i = 0; i < 4; i++) {
4         fb[i].setId(); // Set ID for each employee
5         fb[i].getId(); // Get ID of each employee
6     }
7     return 0;
8 }
9
```

4. Key Points:

- Arrays are useful for handling multiple objects efficiently.
- Individual creation of objects is possible but not scalable for many objects.

5. Output:

- Displays the `id` of each employee after input.

2. Passing Objects as Function Arguments

1. Definition:

- Objects can be passed as function arguments to perform operations using their data members.

2. Example:

```
1 class Complex {
2     int a;
3     int b;
```

```

4 public:
5     void setData(int v1, int v2) {
6         a = v1;
7         b = v2;
8     }
9     void setDataBySum(Complex o1, Complex o2) {
10        a = o1.a + o2.a;
11        b = o1.b + o2.b;
12    }
13    void printNumber() {
14        cout << "Your complex number is " << a << " + " << b << "i" << endl;
15    }
16 };
17

```

3. Usage in main:

```

1 int main() {
2     Complex c1, c2, c3;
3     c1.setData(1, 2); // Set data for c1
4     c1.printNumber();
5     c2.setData(3, 4); // Set data for c2
6     c2.printNumber();
7     c3.setDataBySum(c1, c2); // Add c1 and c2, store result in c3
8     c3.printNumber();
9     return 0;
10 }
11

```

4. Key Points:

- `setDataBySum` uses two objects to compute the sum and assign values to the current object.
- Useful for operations involving multiple objects.

5. Output:

- Displays the sum of two complex numbers in the format `a + bi`.

Short Notes

1. Array of Objects:

- Declared like normal arrays but stores objects: `ClassName arrayName[size];`.
- Example:

```

1 Employee fb[4]; // Array of Employee objects
2

```

- Use loops for efficient operations on all objects.

2. Functions in Array of Objects:

- `setId()`: Assigns `id` and a fixed `salary` to the object.
- `getId()`: Displays the `id` of the object.

3. Passing Objects as Arguments:

- Objects can be passed to functions to perform operations on their data.
- Example: Adding data members of two objects and storing the result in another.

```

1 void setDataBySum(Complex o1, Complex o2) {
2     a = o1.a + o2.a;
3     b = o1.b + o2.b;

```

```
4 }  
5
```

- Functions:

- `setData` : Assigns values to private data members.
- `setDataBySum` : Adds data from two objects.
- `printNumber` : Displays object data.

4. **Syntax Highlights:**

- Array of objects:

```
1 ClassName arrayName[size];  
2
```

- Passing objects:

```
1 void functionName(ClassName obj);
```