Tutorial 23 - Objects Memory Allocation & Using Arrays in Classes in C++

Objects Memory Allocation in C++

1. **Variables**:
   - Memory is allocated only when an object is created.
   - Each object has its own copy of the class variables, allowing different values for each object.
2. **Functions**:
   - Memory for functions is allocated once when the class is declared.
   - All objects share the same copy of the class's functions.

Arrays in Classes

- Arrays store multiple values of the same type in sequential order.
- Useful for managing multiple variables without declaring them individually.

Example: Shop Class

```
class Shop {
    int itemId[100];    // Array to store item IDs
    int itemPrice[100]; // Array to store item prices
    int counter;        // Keeps track of the number of items
public:
    void initCounter(void) { counter = 0; } // Initializes counter to 0
    void setPrice(void);                    // Sets item ID and price
    void displayPrice(void);                // Displays all items and their prices
};

```

Function Definitions

1. **Set Price Function**:
   - Takes input for `itemId` and `itemPrice`.
   - Increments `counter` to add the next item.

```
void Shop::setPrice(void) {
    cout << "Enter ID of your item no " << counter + 1 << endl;
    cin >> itemId[counter];
    cout << "Enter Price of your item" << endl;
    cin >> itemPrice[counter];
    counter++;
}

```

2. **Display Price Function**:
   - Prints all item IDs and prices stored in arrays.

```
void Shop::displayPrice(void) {
    for (int i = 0; i < counter; i++) {
        cout << "The Price of item with ID " << itemId[i] << " is " << itemPrice[i] << endl;
    }
}

```

Main Function

- Demonstrates usage of the `Shop` class.
- Calls `setPrice` multiple times to input item data.
- Displays the entered data using `displayPrice`.

```
int main() {
    Shop dukaan;         // Create an object of Shop
    dukaan.initCounter(); // Initialize counter to 0
    dukaan.setPrice();    // Input first item's data
    dukaan.setPrice();    // Input second item's data
    dukaan.setPrice();    // Input third item's data
    dukaan.displayPrice(); // Display all items and prices
    return 0;
}

```

Program Output

For the following input:

- **Item 1**: ID = 1001, Price = 12
- **Item 2**: ID = 1002, Price = 23
- **Item 3**: ID = 1003, Price = 34

The output is:

```
The Price of item with ID 1001 is 12
The Price of item with ID 1002 is 23
The Price of item with ID 1003 is 34

```

Short Notes

1. **Memory Allocation**:
    - **Variables**: Allocated when objects are created; unique for each object.
    - **Functions**: Allocated once during class declaration; shared by all objects.
2. **Arrays in Classes**:
    - Store multiple values of the same type in sequential order.
    - Manage data efficiently without multiple variable declarations.
3. **Shop Class Example**:
    - **Variables**:
        - `itemId[100]`: Array for item IDs.
        - `itemPrice[100]`: Array for item prices.
        - `counter`: Tracks the number of items.
    - **Functions**:

      - `initCounter`: Initializes the counter to 0.
      - `setPrice`: Inputs item ID and price.
      - `displayPrice`: Displays all items and prices.

4. **Key Syntax**:
   - Arrays are used for storing multiple data points.
   - Loops are used for iterating through array elements.