Aavani Rajesh Perumbessi

Mark zuckerburg

C++ Assignmnet

Introduction to Dynamic Memory Management

- Definition: Allocating memory during runtime using operators.
- Why it's needed:
- Efficient memory usage
- Flexibility in program design
- Operators: new and delete

New Operator

Purpose: Allocates memory dynamically.

Syntax:
int* ptr = new int;

Features:
Returns a pointer to the allocated memory.
Initializes objects (if needed).

Example:
int* array = new int[5];

Delete Operator

Purpose: Frees up memory allocated using new.

Syntax
delete ptr;
delete[] array;

Importance : Prevents memory leaks.

Example
delete ptr;

Class

- A class is a user-defined data type that acts as a blueprint for creating objects.
- It encapsulates data members (attributes) and member functions (methods) to define an entity’s properties and behavior.
- Provides abstraction, encapsulation, and reusability in programming.

Object

- An object is an instance of a class, representing a real-world entity.
- Objects hold specific values for the attributes defined in the class and can perform operations using member functions.

Structure of a Class

```
class ClassName {
 private: // Access specifier (can also be public or protected)
 int data; // Data member
 public:
 void setData(int value) { // Member function
 data = value;
 }
 int getData() {
 return data;
 }
};
```

Creating and Using Objects

```
int main() {
ClassName obj;  // Create an object
obj.setData(42);  // Call member function
cout << obj.getData();  // Access data through method
 return 0; }
```

Constructors

Definition: Special functions for initializing objects.

Types:
Default Constructor
Parameterized Constructor
Copy Constructor

Example:

```
class Example {
    Example() { cout << "Constructor called"; }
};
```

Thank You