
1- Create a C++ program to define a class called Student with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted.

Output

```
Object Created
Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Student {
public:
    Student() {
        cout << "Object Created" << endl;
    }

    ~Student() {
        cout << "Object Deleted" << endl;
    }
};

int main() {
    Student* s = new Student;
    delete s;
    return 0;
}
```

2- Create a C++ program to define a class called Book with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted.

Output

```
Object Created
Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Book {
public:
    Book() {
        cout << "Object Created" << endl;
    }

    ~Book() {
        cout << "Object Deleted" << endl;
    }
};

int main() {
    Book* b = new Book;
    delete b;
    return 0;
}
```

3- Create a C++ program to define a class called Computer with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted.

Output

```
Computer Object Created
Computer Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Computer {
public:
    Computer() {
        cout << "Computer Object Created" << endl;
    }

    ~Computer() {
        cout << "Computer Object Deleted" << endl;
    }
};

int main() {
    Computer* comp = new Computer;
    delete comp;
    return 0;
}
```

4- Create a C++ program to define a class called Student with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted. Also, get data for the student from the user.

Input

```
Student Object Created
Enter student name: amr
Enter student age: 22
```

Output

```
Student Name: amr
Student Age: 22
Student Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Student {
private:
    string name;
    int age;

public:
    Student() {
        cout << "Student Object Created" << endl;
    }

    ~Student() {
        cout << "Student Object Deleted" << endl;
    }

    void getData() {
        cout << "Enter student name: ";
        cin >> name;

        cout << "Enter student age: ";
        cin >> age;
    }

    void displayData() {
        cout << "Student Name: " << name << endl;
        cout << "Student Age: " << age << endl;
    }
};

int main() {
    Student* s = new Student;
    s->getData();
    s->displayData();
    delete s;
    return 0;
}
```

5- Create a C++ program to define a class called Employee with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted. Also, get data for the employee from the user.

Input

```
Employee Object Created
Enter employee name: Ahmed
Enter employee salary: 50000
```

Output

```
Employee Name: Ahmed
Employee Salary: 50000
Employee Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Employee {
private:
    string name;
    int salary;

public:
    Employee() {
        cout << "Employee Object Created" << endl;
    }

    ~Employee() {
        cout << "Employee Object Deleted" << endl;
    }

    void getData() {
        cout << "Enter employee name: ";
        cin >> name;

        cout << "Enter employee salary: ";
        cin >> salary;
    }

    void displayData() {
        cout << "Employee Name: " << name << endl;
        cout << "Employee Salary: " << salary << endl;
    }
};

int main() {
    Employee* e = new Employee;
    e->getData();
    e->displayData();
    delete e;
    return 0;
}
```

6- Create a C++ program to define a class called Car with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted. Also, get data for the car from the user.

Input

```
Car Object Created
Enter car brand: Toyota
Enter car model: carmy
```

Output

```
Car Brand: Toyota
Car Model: carmy
Car Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Car {
private:
    string brand;
    string model;

public:
    Car() {
        cout << "Car Object Created" << endl;
    }

    ~Car() {
        cout << "Car Object Deleted" << endl;
    }

    void getData() {
        cout << "Enter car brand: ";
        cin >> brand;

        cout << "Enter car model: ";
        cin >> model;
    }

    void displayData() {
        cout << "Car Brand: " << brand << endl;
        cout << "Car Model: " << model << endl;
    }
};

int main() {
    Car* c = new Car;
    c->getData();
    c->displayData();
    delete c;
    return 0;
}
```

7- Create a C++ program to define a class called Computer with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted. Also, get data for the computer from the user.

Input

```
Computer Object Created
Enter computer brand: Dell
Enter computer model: XPS13
```

Output

```
Computer Brand: Dell
Computer Model: XPS13
Computer Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class Computer {
private:
    string brand;
    string model;

public:
    Computer() {
        cout << "Computer Object Created" << endl;
    }

    ~Computer() {
        cout << "Computer Object Deleted" << endl;
    }

    void getData() {
        cout << "Enter computer brand: ";
        cin >> brand;

        cout << "Enter computer model: ";
        cin >> model;
    }

    void displayData() {
        cout << "Computer Brand: " << brand << endl;
        cout << "Computer Model: " << model << endl;
    }
};

int main() {
    Computer* comp = new Computer;
    comp->getData();
    comp->displayData();
    delete comp;
    return 0;
}
```

8- Write a C++ program to create a class with a dynamic member using new and print its value.

Output

```
Value: 42
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class DynamicValue {
private:
    int* value;

public:
    DynamicValue() {
        value = new int; // Dynamically allocate memory for an integer in the constructor
    }

    ~DynamicValue() {
        delete value; // Deallocate the memory in the destructor
    }

    void setValue(int num) {
        *value = num; // Set the value of the dynamically allocated integer
    }

    void printValue() {
        cout << "Value: " << *value << endl; // Print the value of the dynamically allocated integer
    }
};

int main() {
    DynamicValue obj;
    obj.setValue(42);
    obj.printValue();

    return 0;
}
```

9- Write a C++ program to create a class with a dynamic array of integers using new and print its values.

Input

```
Enter the size of the array: 4
Enter 4 integers for the array:
1 2 3 4
```

Output

```
Array values:
1 2 3 4
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class DynamicArray {
private:
    int* arr;
    int size;

public:
    DynamicArray(int n) {
        size = n;
        arr = new int[size]; // Dynamically allocate memory for an array of integers in the
constructor
    }

    ~DynamicArray() {
        delete[] arr; // Deallocate the memory in the destructor
    }

    void setValues() {
        cout << "Enter " << size << " integers for the array:" << endl;
        for (int i = 0; i < size; ++i) {
            cin >> arr[i]; // Input values into the dynamically allocated array
        }
    }

    void printValues() {
        cout << "Array values:" << endl;
        for (int i = 0; i < size; ++i) {
            cout << arr[i] << " "; // Output values of the dynamically allocated array
        }
        cout << endl;
    }
};

int main() {
    int arraySize;
    cout << "Enter the size of the array: ";
    cin >> arraySize;

    DynamicArray obj(arraySize);
    obj.setValues();
    obj.printValues();

    return 0;
}
```

10- Create a C++ program to define a class called MobilePhone with a constructor and destructor. Display a message inside the constructor and destructor to indicate when an object is created and deleted.

Output

```
MobilePhone Object Created
MobilePhone Object Deleted
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class MobilePhone {
public:
    MobilePhone() {
        cout << "MobilePhone Object Created" << endl;
    }

    ~MobilePhone() {
        cout << "MobilePhone Object Deleted" << endl;
    }
};

int main() {
    MobilePhone* phone = new MobilePhone;
    delete phone;
    return 0;
}
```
