



ITM SKILLS
UNIVERSITY

**INSTITUTE OF TECHNOLOGY AND MANAGEMENT
SKILLS UNIVERSITY,
KHARGHAR, NAVI MUMBAI**

C++ PROGRAMMING LAB



Prepared by:

Name of Student: _Shikha singh_____

Roll No: _25_____

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Exp. No	List of Experiment
	Write a program to find the roots of a quadratic equation.
	Write a program to calculate the power of a number using a loop.
	Write a program to check if a given string, is a palindrome.
	Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.
	Write a program that finds the largest among three numbers using nested if-else statements
	Write a program that determines the grade of a student based on their marks of 5 subjects using if-else-if ladder.
	Write a program to find the sum of digits of a number until it becomes a single-digit number.
	Write a program to print a Pascal's triangle using nested loops.
	Write a program to calculate the sum of series $1/1! + 2/2! + 3/3! + \dots + N/N!$ using nested loops.
0	Write a program to create an array of strings and display them in alphabetical order.
1	Write a program that checks if an array is sorted in ascending order.
2	Write a program to calculate the sum of elements in each row of a matrix.
3	Write a program to generate all possible permutations of a string.
4	<p>Create a C++ program to print the following pattern:</p> <pre> ***** * * * * * * * * ***** </pre>
5	<p>Write a C++ program to display the following pattern:</p> <pre> 1 232 </pre>

	34543 4567654 34543 232
6	<p>Write a program to creating an inventory management system for a small store. The system should use object-oriented principles in C++. Your program should have the following features:</p> <ul style="list-style-type: none"> • Create a Product class that represents a product in the inventory. Each Product object should have the following attributes: <ul style="list-style-type: none"> • Product ID (an integer) • Product Name (a string) • Price (a floating-point number) • Quantity in stock (an integer) • Implement a parameterized constructor for the Product class to initialize the attributes when a new product is added to the inventory.
7	Write a program to manage student records. Create a class Student with attributes such as name, roll number, and marks. Implement methods for displaying student details, adding new students, and calculating the average marks of all students in the record system.
8	Write a program that implements a basic calculator. Use a class Calculator with methods to perform addition, subtraction, multiplication, and division of two numbers. The program should allow the user to input two numbers and select an operation to perform.
9	Write a program to simulate a simple online shop. Create a class Product with attributes like name, price, and quantity in stock. Implement methods for adding products to the shopping cart, calculating the total cost, and displaying the contents of the cart.
0	Write a program to manage student grades for a classroom. Create a class Student with attributes for student name and an array to store grades. Implement methods for adding grades, calculating the average grade, and displaying the student's name and grades. Use constructors and destructors to initialize and release resources.

Name of Student: _Shikha singh_____

Roll Number: __25_____

Experiment No: 19

Title: 19. Write a program to simulate a simple online shop. Create a class Product with

attributes like name, price, and quantity in stock. Implement methods for adding products to the shopping cart, calculating the total cost, and displaying the contents of the cart.

Theory:

- **Classes and Inheritance:**
 - You've defined a base class Product with common attributes such as productID, name, and price.
 - Derived classes (Electronics, Clothing, Books) inherit from the Product class, showcasing the "is-a" relationship.
- **Encapsulation:**
 - You've encapsulated the data members in the Product class, making them private. Access to these members is provided through public member functions, following the principle of encapsulation.
- **Polymorphism:**
 - Though not explicitly shown in this code, polymorphism could be implemented by having virtual functions in the base class and overriding them in the derived classes.
- **User Interaction:**
 - The program interacts with the user through a menu-driven system, accepting user choices and performing actions accordingly.
- **Dynamic Memory:**
 - Your program uses static objects for products, but for a more scalable solution, dynamic memory allocation (e.g., vectors or dynamic arrays) could be considered.

Code:

```
#include <iostream>
#include <vector>
using namespace std;
```

```

// Base class for products
class Product {
public:
    int productID;
    string name;
    double price;
    Product(int id, string productName, double productPrice) {
        productID = id;
        name = productName;
        price = productPrice;
    }
};

// Derived class for Electronics
class Electronics : public Product {
public:
    Electronics(int id, string name, double price) : Product(id, name, price) {
    }
};

// Derived class for Clothing
class Clothing : public Product {
public:
    Clothing(int id, string name, double price) : Product(id, name, price) {
    }
};

// Derived class for Books
class Books : public Product {
public:
    Books(int id, string name, double price) : Product(id, name, price) {
    }
};

// Shopping cart to store products
class ShoppingCart {
public:
    vector<Product> cart;
    void addToCart(Product product) {
        cart.push_back(product);
    }
    double calculateTotalPrice() {
        double total = 0;
        for (const Product &product : cart) {
            total += product.price;
        }
        return total;
    }
    void displayCart() {
        if (cart.empty()) {
            cout << "Your shopping cart is empty." << endl;
        } else {
            cout << "Items in your shopping cart:" << endl;
            for (const Product &product : cart) {
                cout << "Product ID: " << product.productID << ", Name: " << product.name << ", Price: " <<
product.price << endl;
            }
        }
    }
};

```

```

    }
    cout << "Total Price: $" << calculateTotalPrice() << endl;
}
};

int main() {
    // Create some sample products
    Electronics laptop(1, "Laptop", 800.0);
    Clothing shirt(2, "T-Shirt", 20.0);
    Books novel(3, "Great Novel", 15.0);
    // Initialize the shopping cart
    ShoppingCart cart;
    int choice;
    do {
        cout << "\nMenu:\n";
        cout << "1. Add Laptop to Cart\n";
        cout << "2. Add T-Shirt to Cart\n";
        cout << "3. Add Great Novel to Cart\n";
        cout << "4. Display Cart\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
            case 1:
                cart.addToCart(laptop);
                cout << "Laptop added to cart." << endl;
                break;
            case 2:
                cart.addToCart(shirt);
                cout << "T-Shirt added to cart." << endl;
                break;
            case 3:
                cart.addToCart(novel);
                cout << "Great Novel added to cart." << endl;
                break;
            case 4:
                cart.displayCart();
                break;
            case 5:
                cout << "Thank you for shopping. Goodbye!" << endl;
                break;
            default:
                cout << "Invalid choice. Please try again." << endl;
        }
    } while (choice != 5);
    return 0;
}

```

Output: (screenshot)

```
Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 1
Laptop added to cart.

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 2
T-Shirt added to cart.

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 4
Items in your shopping cart:
Product ID: 1, Name: Laptop, Price: 800
Product ID: 2, Name: T-Shirt, Price: 20
Total Price: $820

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 5
Thank you for shopping. Goodbye!
shikhasingh@SHIKHAS-MacBook-Air C++ %
```

Ln 1, Col 1 (2968 selected) Spaces: 4 UTF-8 LF {} C++ Mac

Test Case: Any two (screenshot)

```
Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 1
Laptop added to cart.

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 2
T-Shirt added to cart.

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 4
Items in your shopping cart:
Product ID: 1, Name: Laptop, Price: 800
Product ID: 2, Name: T-Shirt, Price: 20
Total Price: $820

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 5
Thank you for shopping. Goodbye!
shikhasingh@SHIKHAS-MacBook-Air C++ %
```

0 Ln 1, Col 1 (2968 selected) Spaces: 4 UTF-8 LF {} C++ Mac


```
Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 1
Laptop added to cart.

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 3
Great Novel added to cart.

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 4
Items in your shopping cart:
Product ID: 1, Name: Laptop, Price: 800
Product ID: 3, Name: Great Novel, Price: 15
Total Price: $815

Menu:
1. Add Laptop to Cart
2. Add T-Shirt to Cart
3. Add Great Novel to Cart
4. Display Cart
5. Exit
Enter your choice: 5
Thank you for shopping. Goodbye!
shikhasingh@SHIKHAS-MacBook-Air C++ %
```

Conclusion:

In conclusion, your program successfully models a simple online shopping scenario using C++ and OOP concepts. The use of classes and inheritance promotes code reusability and maintainability, allowing for easy extension when introducing new product types. The user interface provides a straightforward interaction for adding products to the shopping cart and viewing the cart's contents.