LTCE

TASK

Online Shopping System

Objective: To develop a python program that stimulates an online shopping system by implementing classes for products, customers, and shopping carts. The program should include methods for adding items to the cart, calculating total costs, processing orders, and managing inventory.

Requirements:

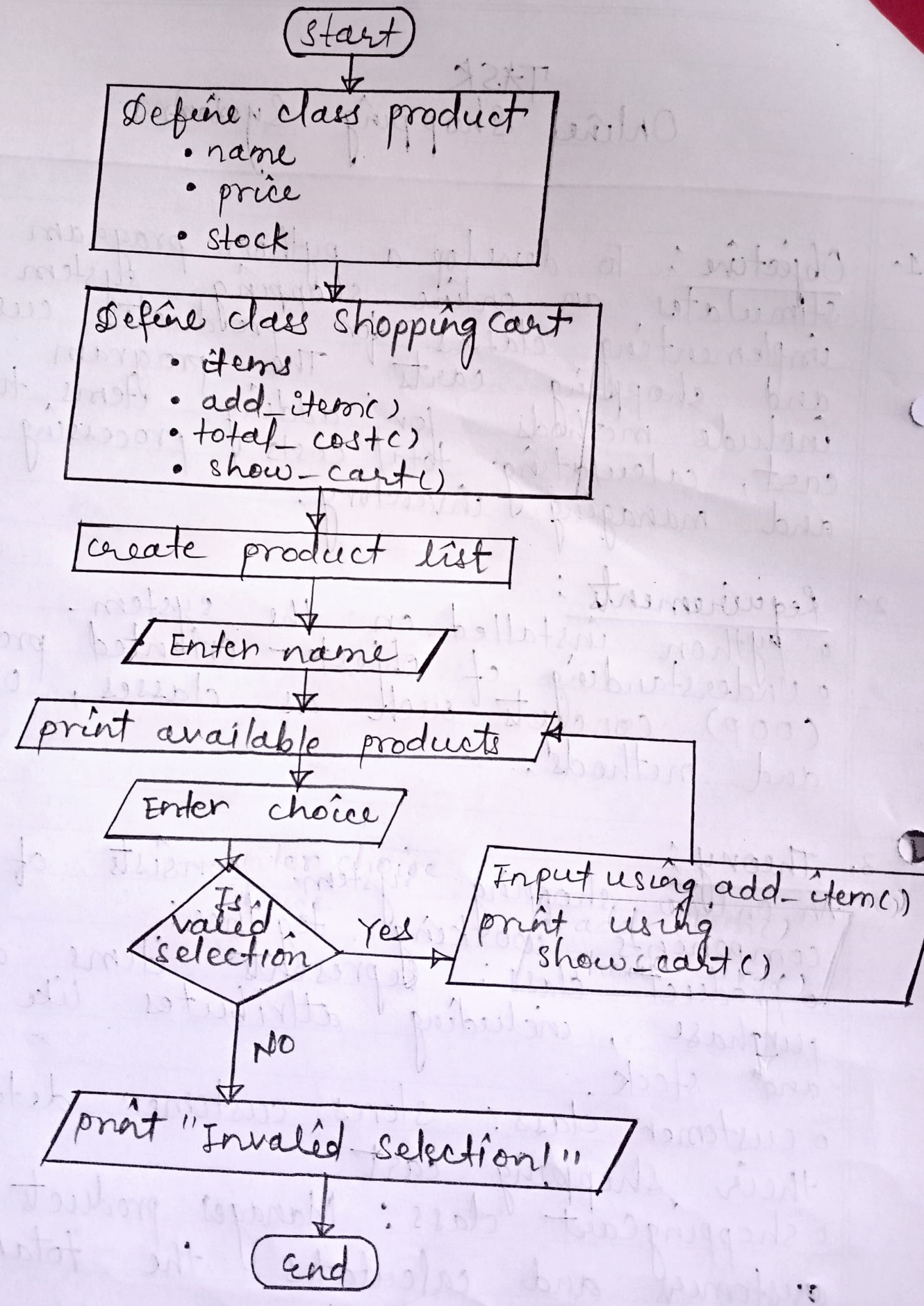
- Python installed on the system.
- Understanding of object-oriented programming (OOP) concepts such as classes, objects, and methods.

Theory:

An online shopping system consists of multiple components working together:

- product class: Represents items available for purchase, including attributes like name, price, and stock.
- customer class: stores customer details and their shopping cart.
- shoppingCart class: Manages products added by customers and calculates the total cost.

Flowchart:


Start
Define class product
• name
• price
• stock
Define class shopping cart
• items
• add_item()
• total_cost()
• show_cart()
create product list
Enter name
print available products
Enter choice
Is valid selection
Yes
Input using add_item()
print using show_cart()
NO
print "Invalid selection!"
end

- Order processing: Handles order placement and stock management.

Algorithm

a. Define a product class with attributes for name price, and stock quantity.
b. Define a customer class to store customer details and an instance of a shopping cart.
c. Create a shopping cart class with methods to add products, remove products, and calculate total cost.
d. Implement an interactive shopping experience where the user can select products, add them to the cart, and proceed to checkout
e. Demonstrate the working system with sample products and customer transactions.

program code:

```
class Product:
    def __init__(self, name, price, stock):
        self.name = name
        self.price = price
        self.stock = stock

    def update_stock(self, quantity):
        if self.stock >= quantity:
            self.stock -= quantity
            return True
        Return False
```

```
class shoppingcart:
    def __init__(self):
        self.items = {}

    def add_item(self, product, quantity):
        if product.update_stock(quantity):
            self.items[product] = self.items.
            get(product, 0) + quantity.
        else:
            print(f"sorry, {product.name} is
            running out fast!")

    def total_cost(self):
        return sum(product.price * qty for
        product, qty in self.items.items())

    def show_cart(self):
        if not self.items:
            print("Your shopping cart is empty!"
        else:
            print("Your shopping cart:")
            for product, qty in self.items.items()

print(f"{product.name}: {qty}")

class customer:
    def __init__(self, name):
        self.name = name
        self.cart = shoppingcart()
```

```
def checkout(self):
    total = self.cart.total_cost()
    print(f"{self.name}, your total amount
    is ₹{total}.")
    self.cart = shoppingCart()

products = [
    product("Laptop", 65000, 5),
    product("phone", 40000, 10),
    product("Headphones", 3000, 15),
    product("smartwatch", 7000, 8),
    product("Bluetooth speaker", 4500, 12)
    ]
customer = customer(input("Enter your name
while True:
    print("\n Available products:")
    for i, product in enumerate(products):
        print(f"{i+1}. {product.name} -
        ₹{product.price} ({product.stock} in
        stock)")

    choice = input("Enter product number
    type 'checkout' to finish:")

    if choice.lower() == "checkout":
        customer.checkout()
        break

If choice.isdigit() and 1 <= int(choice) <=
len(products):
```

```
            quantity = int(input("📦 Enter quantity: "))
customer.cart.add_item(products[int(choice)-1], quantity)
            customer.cart.show_cart()
        else:
            print("⚠ Invalid selection!")
```

conclusion:

The program successfully simulates an interactive online shopping system with product management, shopping cart functionality, and order processing. This demonstrates OOP principles and real-world application of python programming.