**Lab Final Term Paper**

Name : Zeeshan Ali

Roll no : SU92-BSITM-F22-019

**Question no 01 :**

**Code :**

/\*Design a game where different characters (e.g., Player, Enemy) can interact with each other. Implement a base class Character with a virtual method attack(). Derive classes like Player and Enemy from Character, each having its own attack behavior.Use polymorphism to make different characters attack each other.\*/

#include <iostream>

using namespace std;

class Chracter

{

public :

void virtual attack ()

{

cout << "I am a base class attack method.\n\n" << endl;

}

};

class Player : public Chracter

{

void attack ()

{

cout << "\n\n\n--------------Player Attack---------------" << endl;

cout << "I am a player and i am gonna win the game. " << endl;

}

};

class Enemy : public Chracter

{

void attack ()

{

cout << "\n\n\n--------------Enemy Attack---------------" << endl;

cout << "I am an enemy of the player.\nand I am gonna try my best to defeat the player. " << endl;

}

};

int main()

{

int user\_choice ;

do

{

cout << "-----------Game-----------" << endl;

cout << "1.Player Attack functionality." << endl;

cout << "2.Enemy Attack functionality." << endl;

cout << "\nEnter your choice."<< endl;

cin >> user\_choice ;

Chracter\* chrac ;

Player play ;

Enemy Enem ;

switch (user\_choice )

{

case 1 :

chrac = &play;

chrac->attack();

break ;

case 2 :

chrac = &Enem;

chrac->attack();

}

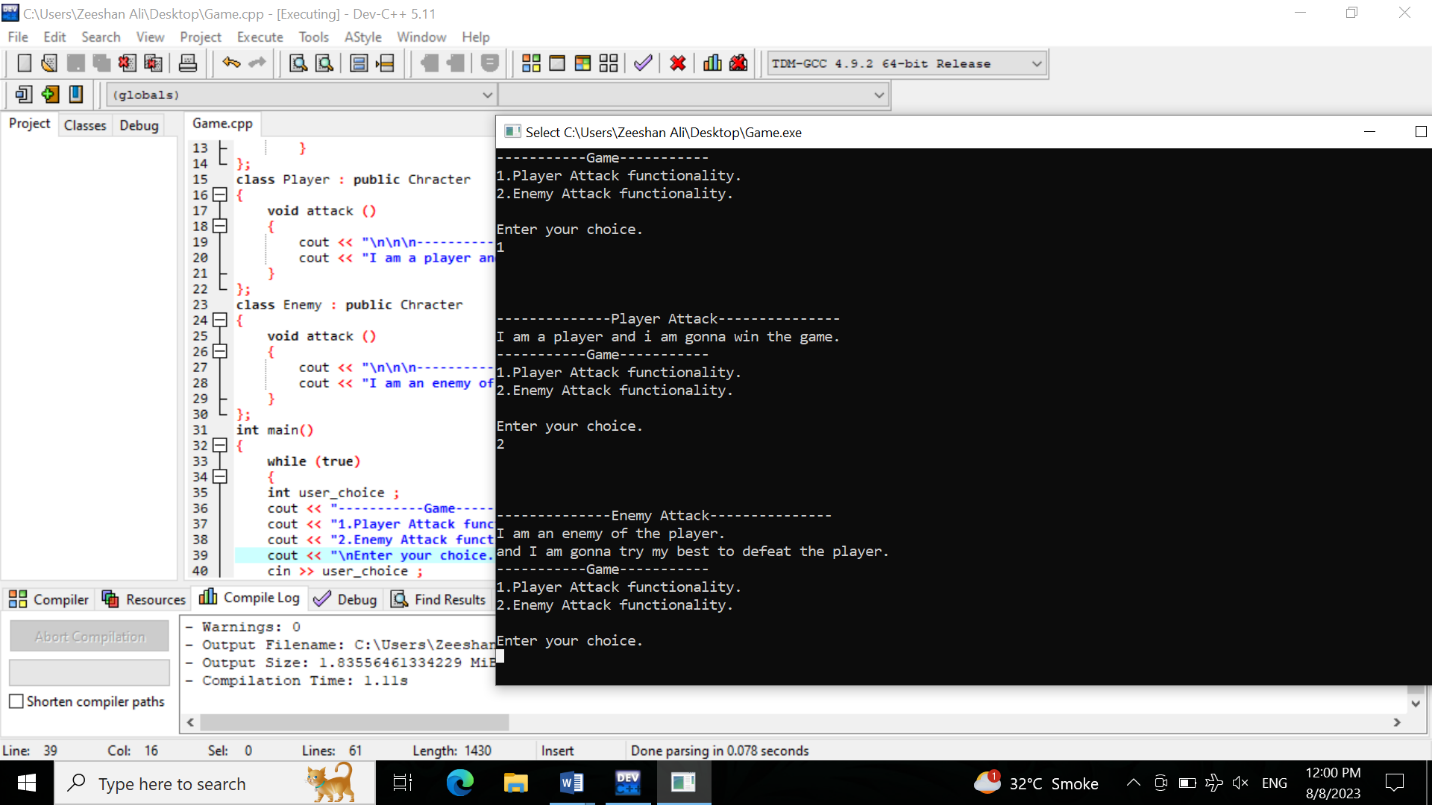
}

while (user\_choice != 2 );

return 0 ;

}

**Output :**

****

**Question no 02 :**

**Code :**

#include <iostream>

#include <string>

using namespace std;

class LeCream {

private:

int price;

string flavour;

int num\_of\_Scoops;

bool hasWafer;

public:

LeCream(int \_price, string \_flavor, int \_scoops) {

price = \_price;

flavour = \_flavor;

num\_of\_Scoops = \_scoops;

hasWafer = false;

}

void add\_Wafer() {

hasWafer = true;

}

int CalculatePrice() {

int totalPrice = num\_of\_Scoops \* price;

if (hasWafer) {

totalPrice += 10;

}

return totalPrice;

}

void displayOrder() {

cout << "\nOrder Summary:" << endl;

cout << "Flavour: " << flavour << endl;

cout << "Number of scoops: " << num\_of\_Scoops << endl;

cout << "Vanilla wafer: ";

if (hasWafer) {

cout << "Yes" << endl;

} else {

cout << "No" << endl;

}

cout << "Total Amount: Rs " << CalculatePrice() << endl;

}

};

int main() {

cout << "Welcome to LeCream Ice Cream Vendor!" << endl;

cout << "=====================================" << endl;

int choice;

do

{

cout << "\nMenu:" << endl;

cout << "1. Flavored Ice Cream" << endl;

cout << "2. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

{

cout << "Flavor Options:" << endl;

cout << "1. Chocolate" << endl;

cout << "2. Vanilla" << endl;

cout << "3. Strawberry" << endl;

cout << "4. Mango" << endl;

cout << "5. Tutti Fruit" << endl;

cout << "6. Almond Crunch" << endl;

cout << "7. Coffee" << endl;

int flavorChoice;

cout << "Enter the flavor choice (1-7): ";

cin >> flavorChoice;

if (flavorChoice < 1 || flavorChoice > 7)

{

cout << "Invalid flavor choice." << endl;

}

int numScoops;

cout << "Enter the number of scoops\nYou must select num of scoops 2 or 3" << endl;

cin >> numScoops;

if (numScoops < 2 || numScoops > 3)

{

cout << "Invalid number of scoops.Exiting." << endl;

}

string flavors[] = {"Chocolate", "Vanilla", "Strawberry", "Mango", "Tutti Fruit", "Almond Crunch", "Coffee"};

int basePrice;

if (flavorChoice == 1)

{

if (numScoops == 2)

{

basePrice = 120;

}

else if (numScoops == 3)

{

basePrice = 180;

}

}

else {

if (numScoops == 2)

{

basePrice = 100;

}

else if (numScoops == 3)

{

basePrice = 150;

}

}

LeCream leCream(basePrice, flavors[flavorChoice - 1], numScoops);

char waferChoice;

cout << "Do you want to add a vanilla wafer?\nEnter Y for Yes and N for No (y/n): ";

cin >> waferChoice;

if (waferChoice == 'y' || waferChoice == 'Y') {

leCream.add\_Wafer();

}

leCream.displayOrder();

break;

}

case 2:

cout << "Thank you for visiting LeCream!" << endl;

break;

default:

cout << "Invalid choice. Please select again." << endl;

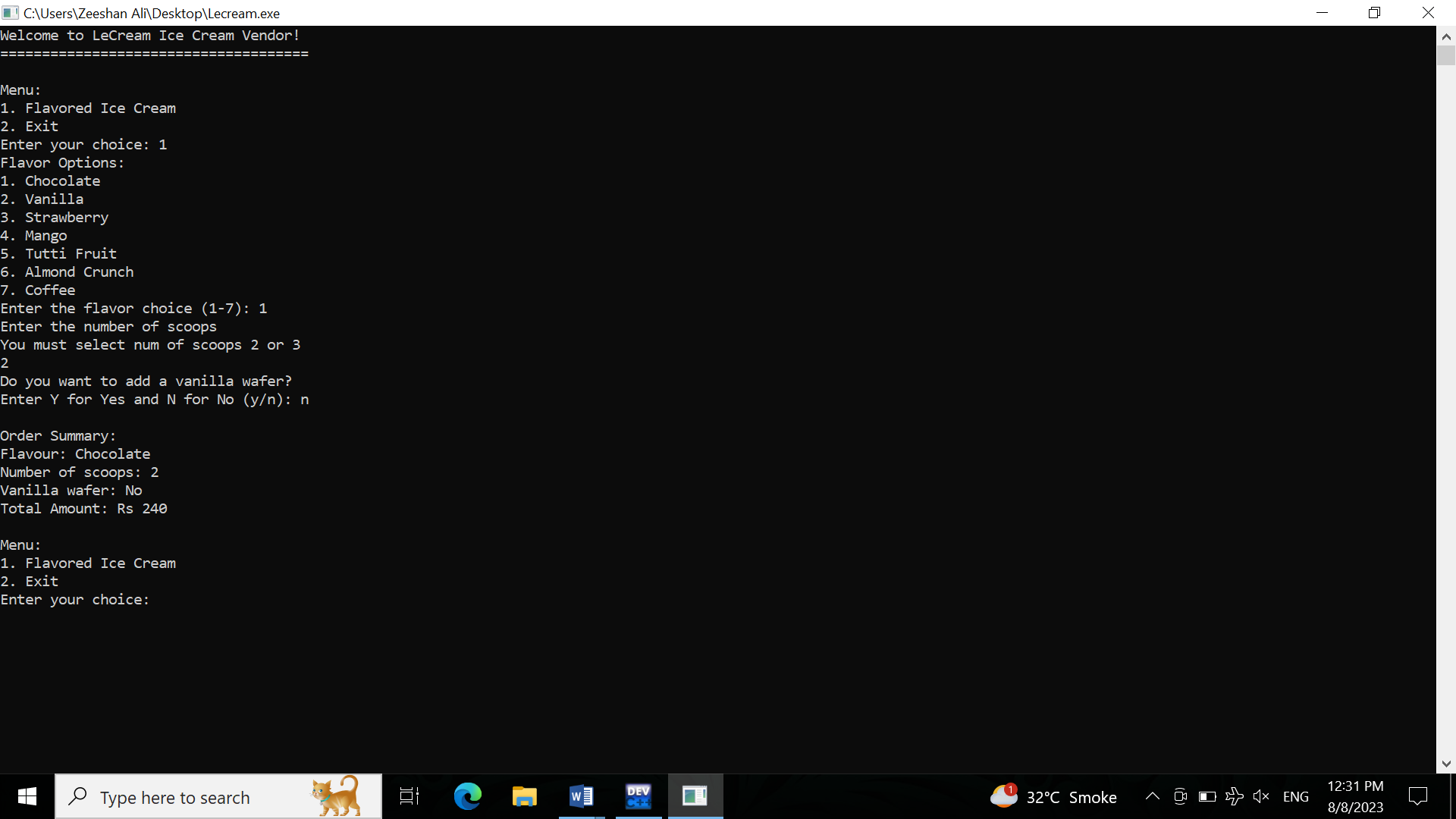
}

} while (choice != 2);

return 0;

}

**Output :**

****