DAY 1 Date:- 08/02/2024

Question 1. Guess the Number Game Implement a "guess the number" game using a while loop. The program should randomly select a number between 1 and 100, and then it prompts the user to guess the number. The loop continues until the user guesses the number correctly, providing hints ("too high" or "too low") after each guess.

PROGRAM

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main() {

srand(time(0));

int secretNumber = rand() % 100 + 1;

int guess;

int attempts = 0;

cout << "Welcome to the Guess the Number game!" << endl;

while (true) {

cout << "Enter your guess (between 1 and 100): ";

cin >> guess;

attempts++;

if (guess == secretNumber) {

cout << "Congratulations! You guessed the number in " << attempts << " attempts." << endl;

break;

} else if (guess < secretNumber) {

cout << "Too low! Try again." << endl;

} else {

cout << "Too high! Try again." << endl;

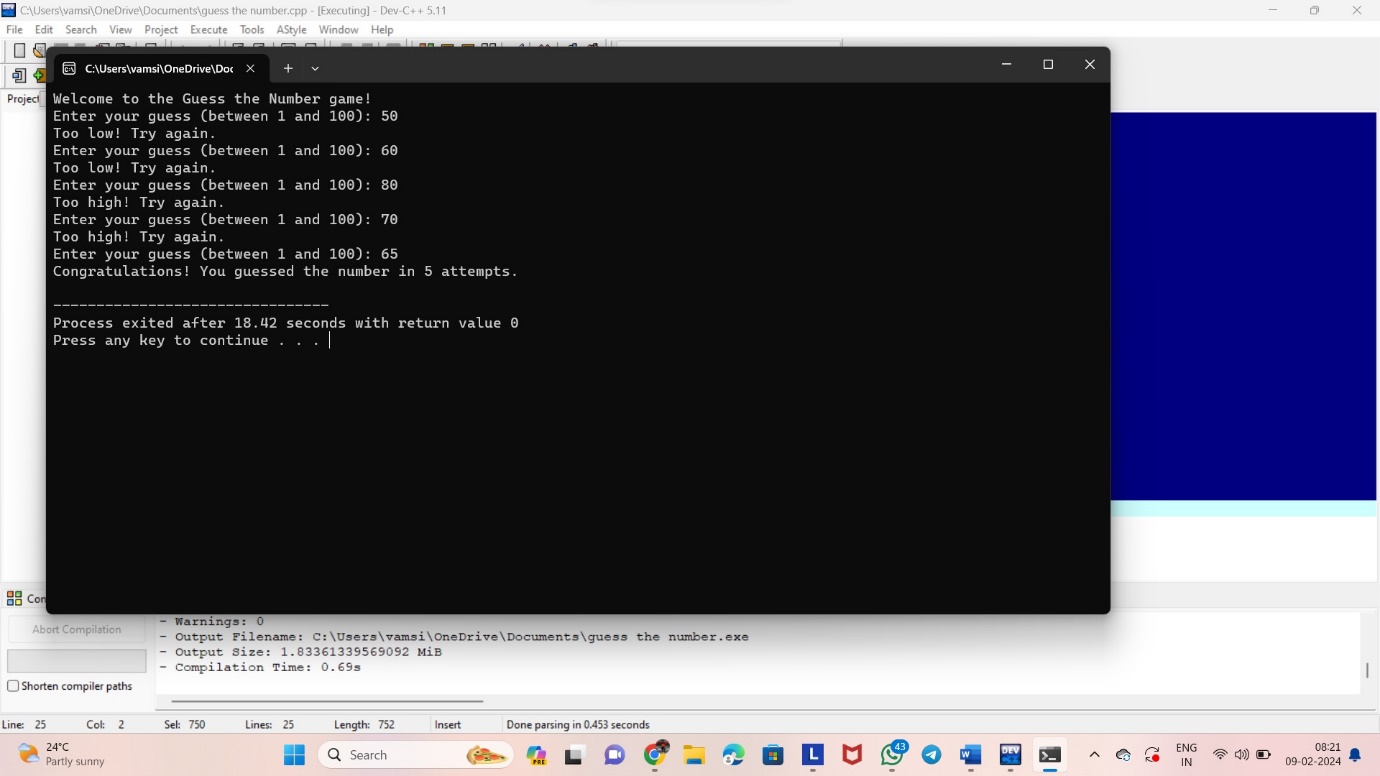
}

}

return 0;

}

OUTPUT



2.Write a program that prompts the user to enter a positive integer. Use a do-while loop to validate the input, ensuring the user re-enters the value until a valid positive integer is provided. Discuss why a do-while loop is more suitable for this scenario compared to a for or while loop.

PROGRAM

#include <iostream>

int main() {

int number;

do {

std::cout << "Enter a positive integer: ";

std::cin >> number;

if (number <= 0) {

std::cout << "Invalid input. Please enter a positive integer." << std::endl;

}

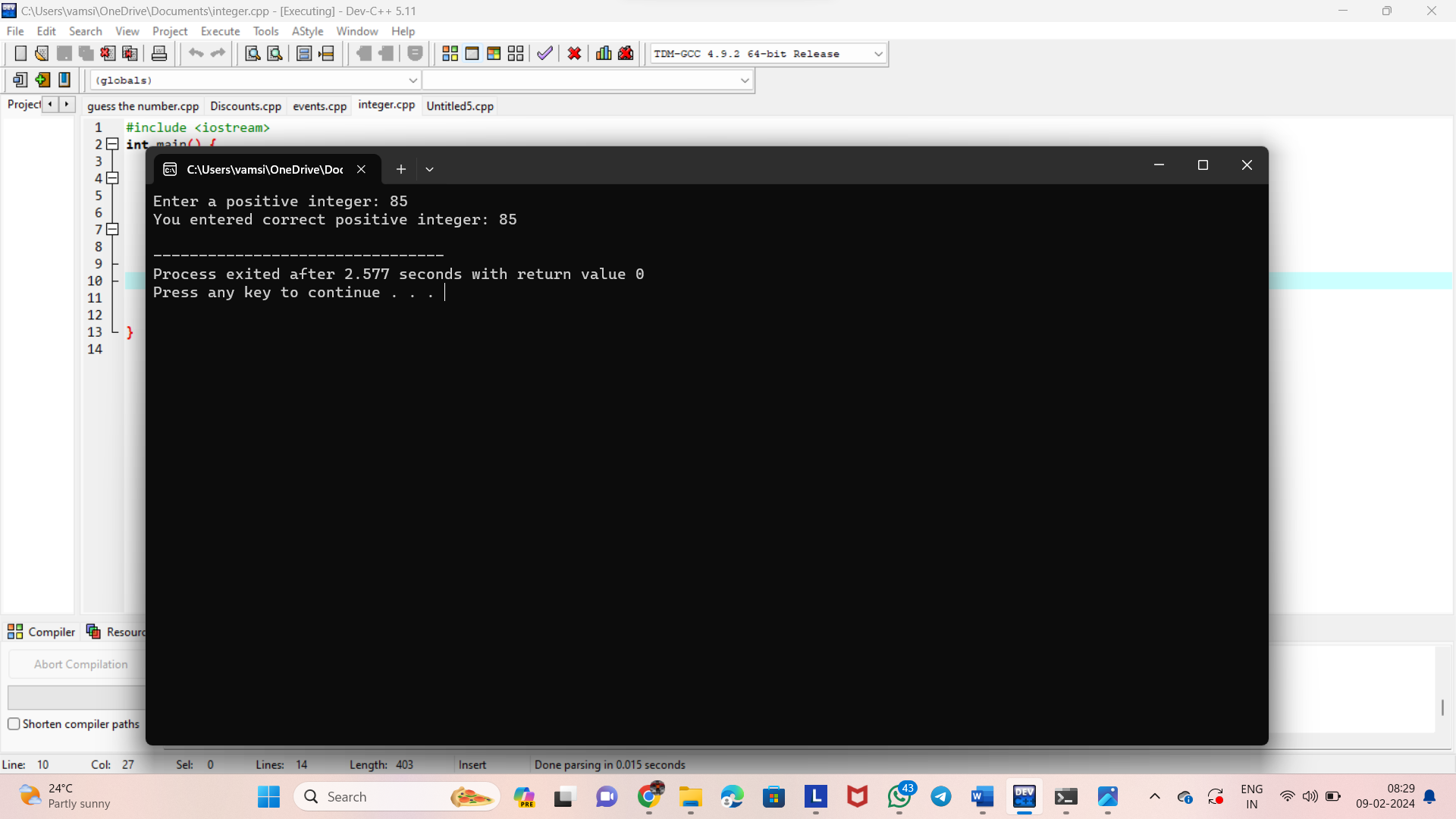
} while (number <= 0);

std::cout << "You entered correct positive integer: " << number << std::endl;

return 0;

}

OUTPUT



3.Background: You are developing a software module for a sports competition where scores are represented as integers. The competition has three events, and each participant can score between 0 to 100 points in each event. Your task is to write a C++ function to determine the highest score among the three events for a given participant, ensuring that your solution is efficient and readable.

PROGRAM

#include<iostream>

using namespace std;

int main(){

int e1,e2,e3;

cout<<"enter scores of three events:"<<endl;

cout<<"event1:";

cin>>e1;

cout<<"event2";

cin>>e2;

cout<<"event3";

cin>>e3;

if(e1>=e2 && e1>=e3){

cout<<"the highest score in event is:"<<e1<<endl;}

else if(e2>=e1 && e2>=e3){

cout<<"the highest score in event is:"<<e2<<endl;}

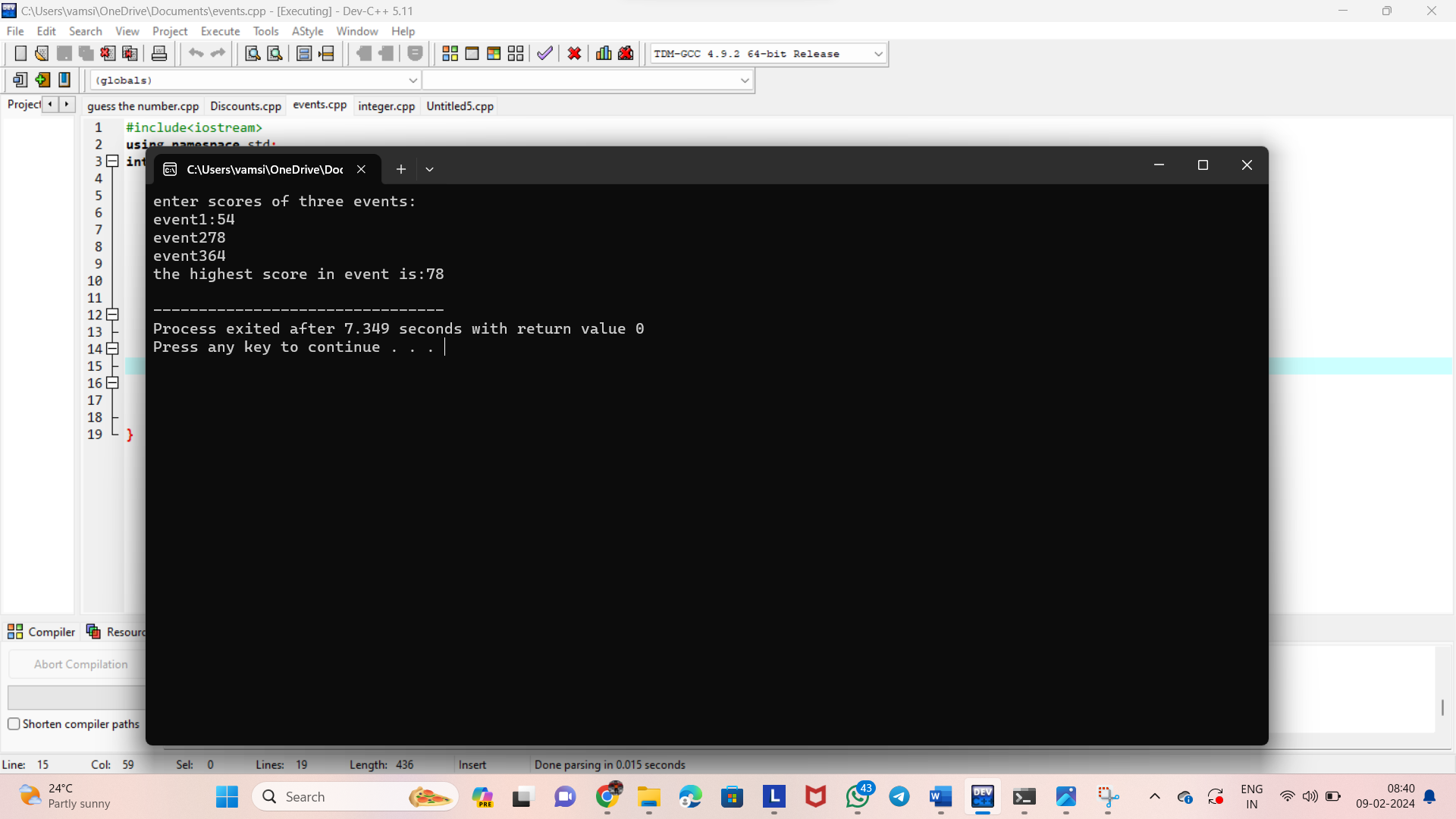
else{

cout<<"the highest scored event is:"<<e3<<endl;

}

}

OUTPUT



4.Background:

You are developing an e-commerce application where you need to apply discounts to purchases. The discount logic varies; it can be a fixed amount off, a percentage of the purchase price, or special rules for bulk purchases. To simplify the implementation, you decide to use function overloading to create several applyDiscount functions, each handling a different type of discount calculation.

Question:

Implement function overloading to handle three types of discounts:

A fixed amount discount off the total purchase price.

A percentage discount off the total purchase price.

A discount for bulk purchases: if the number of items is above a certain threshold, apply a fixed discount per item.

PROGRAM

#include<iostream>

using namespace std;

double applydiscount(double total,double fixedamountdiscount){

return total - fixedamountdiscount;

}

int applydiscount(int total,int percentagediscount){

int discount=(percentagediscount/100)\*total;

return total-discount;

}

int applydiscount(int unitprice,int quantity,int discountperitem,int bulkthreshold)

{

if (quantity>=bulkthreshold){

return ((unitprice-discountperitem)\*quantity);

}

else{

return 0;

}

}

int main(){

double total=2000.0;

double percentagediscount=10;

double fixedamountdiscount=200.0;

int bulkthreshold=10;

int discountperitem=10;

int quantity=11;

int unitprice=100;

cout<<"original purchase total:"<<total<<endl;

double discounttotal1=applydiscount(total,fixedamountdiscount);

cout<<"after discount total amount is:"<<discounttotal1<<endl;

int discounttotal2=applydiscount(total,percentagediscount);

cout<<"after discoount total amount is:"<<discounttotal2<<endl;

int discounttotal3=applydiscount(unitprice,quantity,discountperitem,bulkthreshold);

cout<<"after discount total price is:"<<discounttotal3<<endl;

return 0;

}

OUTPUT

