**Chapter 5**

3)

// Distance Traveled

// Sam Mallet 6/26/2021

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

int speed, time;

cout << "Enter the speed of your vehicle in miles per hour: ";

cin >> speed;

cout << "Enter your hours traveled: ";

cin >> time;

cout << endl << "Hour" << setw(25) << "Miles Traveled" << endl;

cout << setfill('-') << setw(30) << "" << endl;

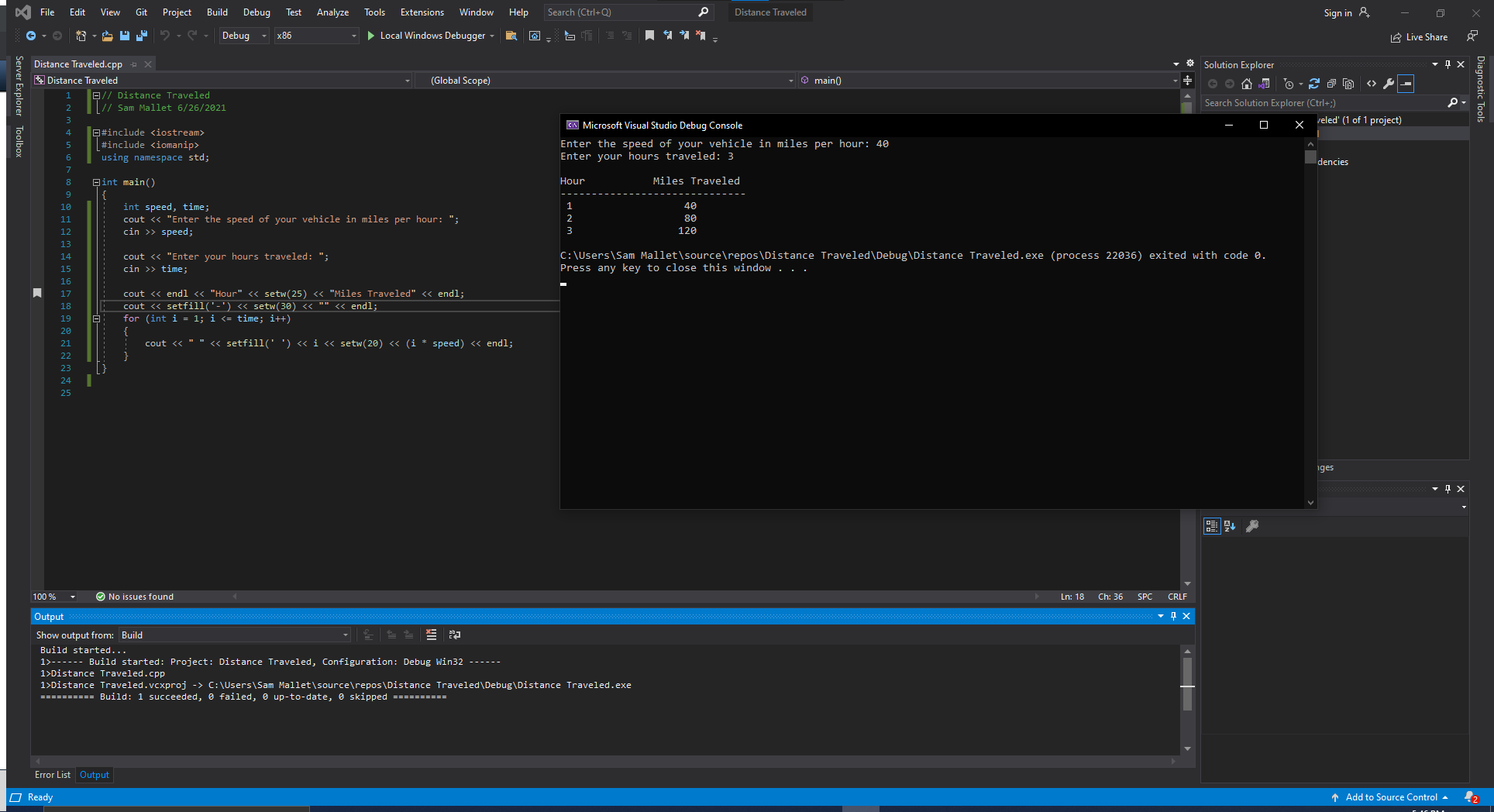
for (int i = 1; i <= time; i++)

{

cout << " " << setfill(' ') << i << setw(20) << (i \* speed) << endl;

}

}



8)

// Pennies for Pay

// Sam Mallet 6/26/2021

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

int time = 0;

double profit = .01;

cout << "Enter the amount of days you worked this month: ";

cin >> time;

while (time < 1 || time > 31)

{

cout << endl << "Out of Range: days worked must be betwee 1 and 31, try again.";

cout << "Enter the amount of days you worked this month: ";

cin >> time;

}

cout << endl << "Day" << setw(25) << "Money Earned" << endl;

cout << setfill('-') << setw(30) << "" << endl;

for (int i = 1; i <= time; i++)

{

cout << " " << setfill(' ') << i << setw(20);

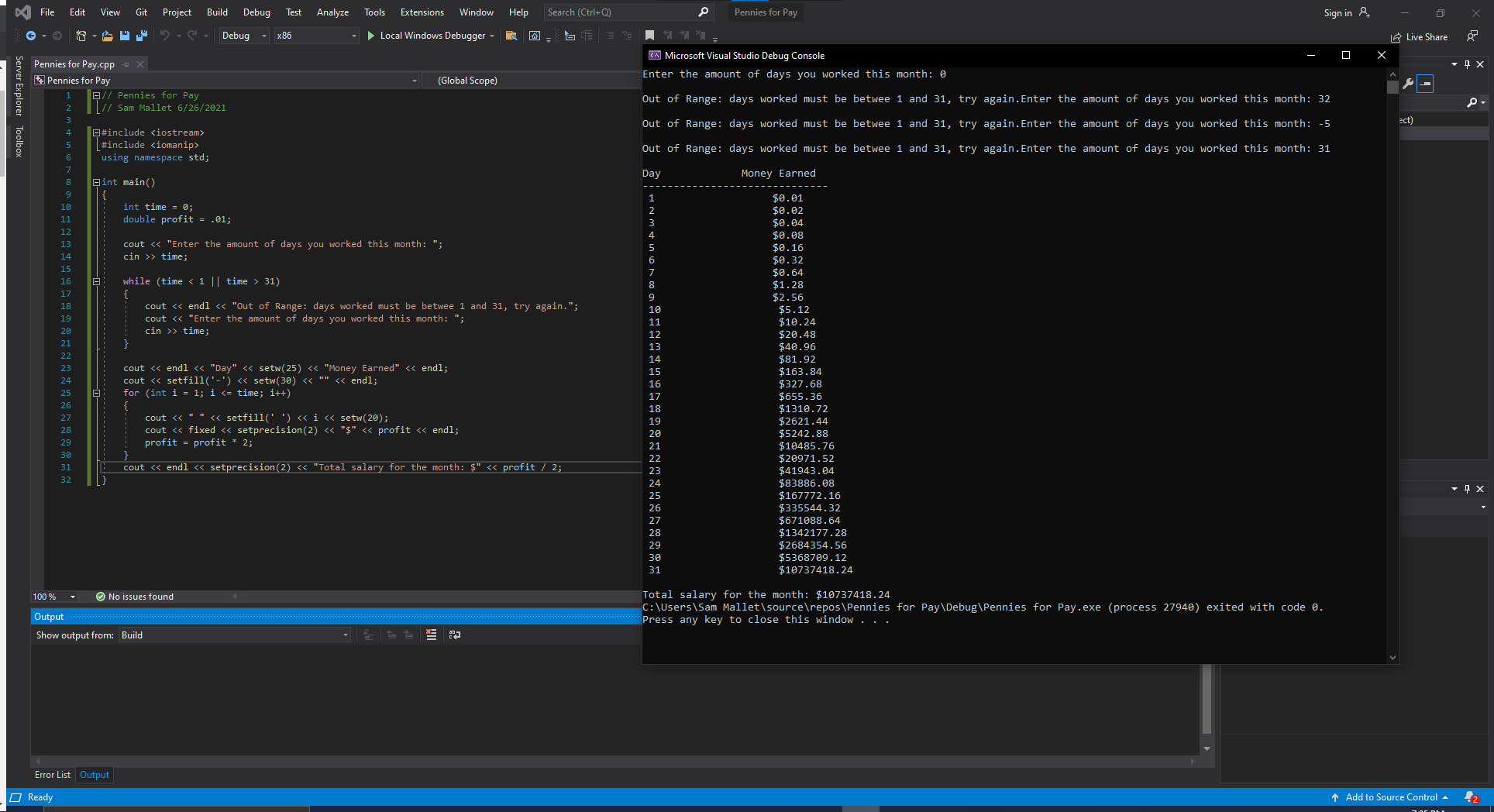
cout << fixed << setprecision(2) << "$" << profit << endl;

profit = profit \* 2;

}

cout << endl << setprecision(2) << "Total salary for the month: $" << profit / 2;

}



12)

// Random Number Guessing Game

// Sam Mallet 6/26/2021

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

unsigned seed = time(0);

srand(seed);

int randNum = rand() % 100 + 1;

int guess;

cout << "I am thinking of a number 1-100. Enter your guess here: ";

cin >> guess;

while (guess != randNum)

{

if (guess < randNum)

{

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

cin >> guess;

}

else if (guess > randNum)

{

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

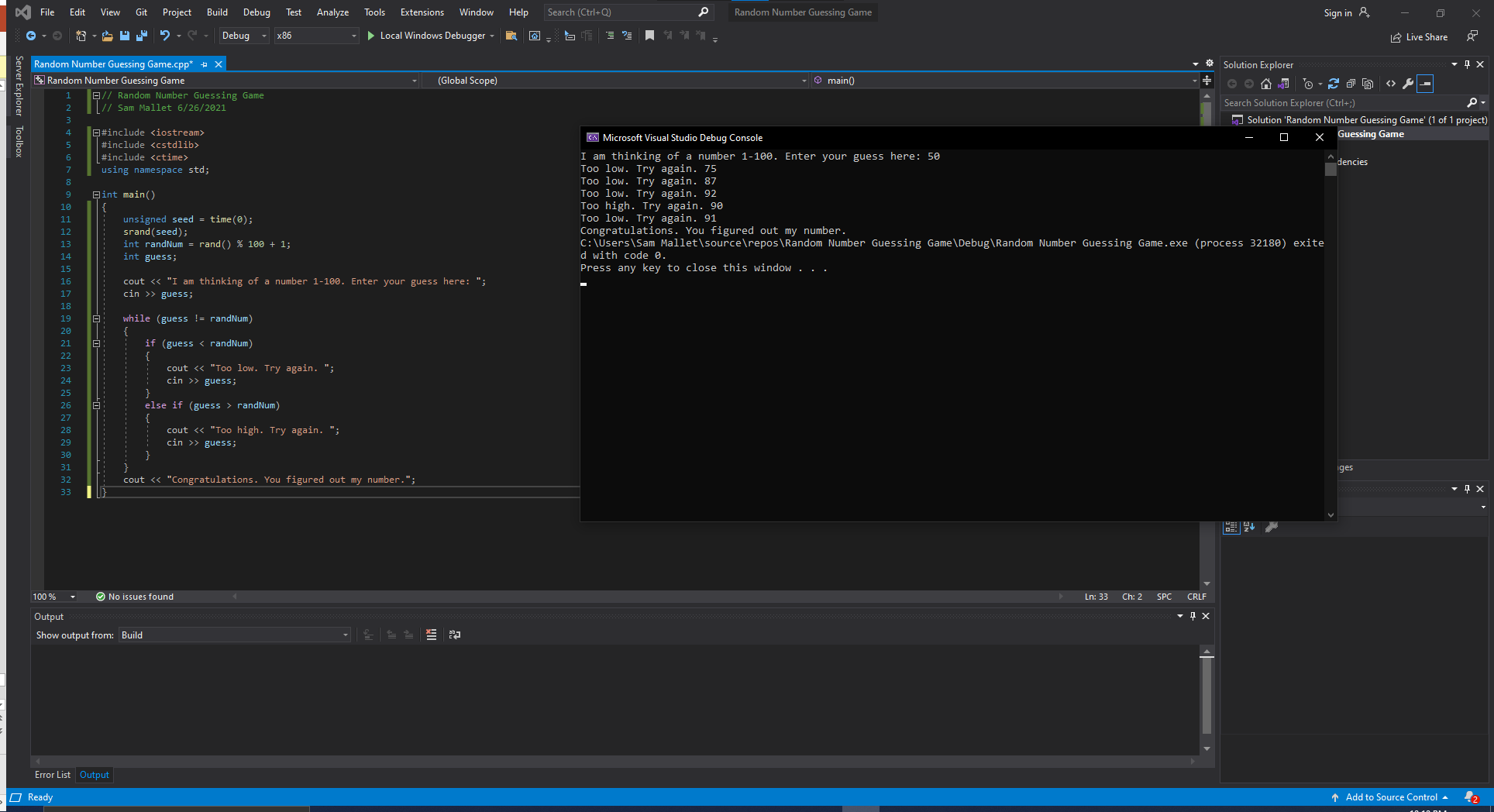
cin >> guess;

}

}

cout << "Congratulations. You figured out my number.";

}



26)

// Total and Average Rainfall

// Sam Mallet 6/26/2021

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

int main()

{

string month\_start, month\_end;

double rain;

double total\_precip = 0;

int total\_rain = 0;

ifstream inFile("C:\\Users\\Sam Mallet\\source\\repos\\rainfall.txt");

inFile >> month\_start >> month\_end;

while (inFile >> rain)

{

total\_precip += rain;

total\_rain++;

}

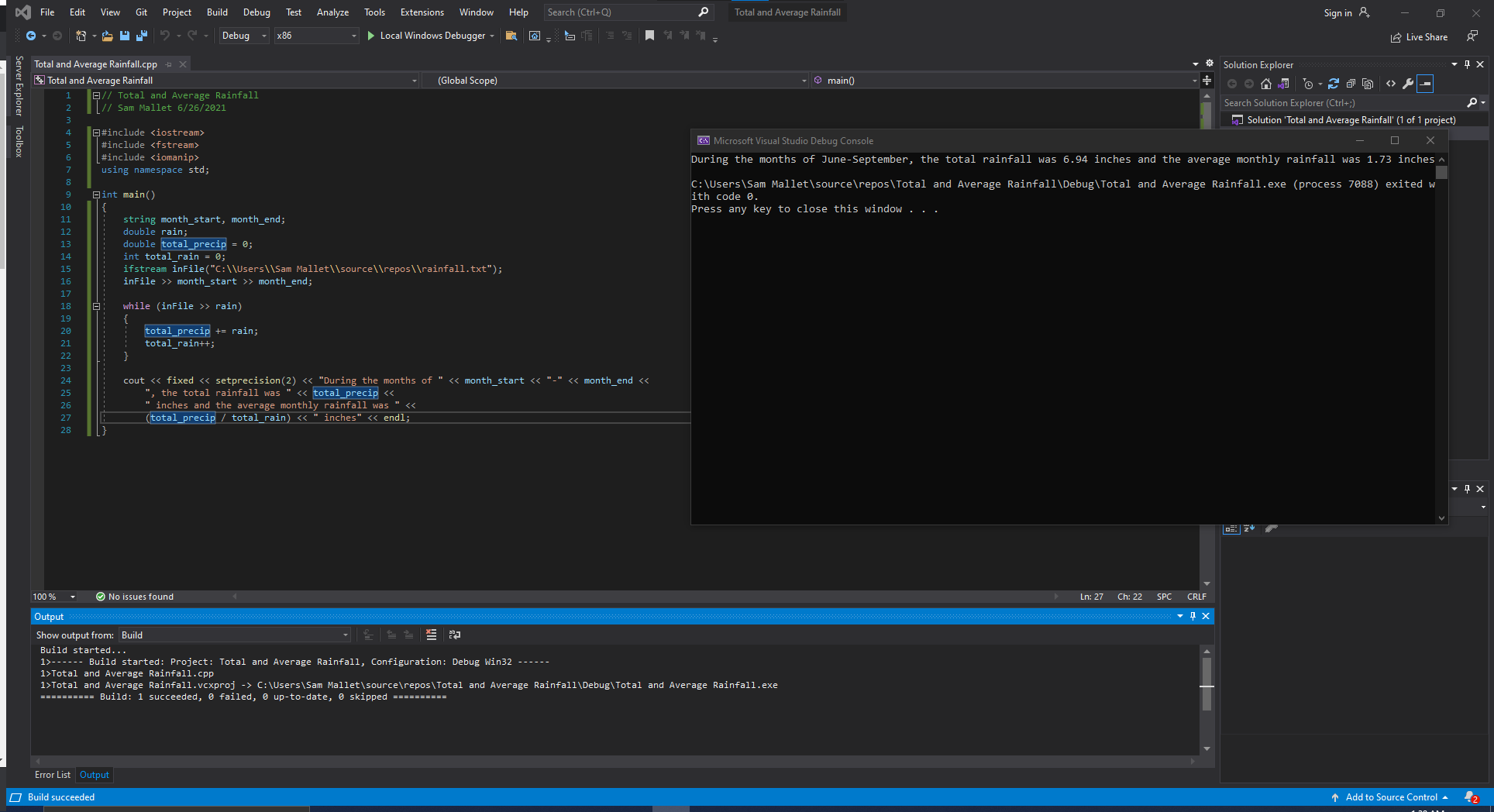
cout << fixed << setprecision(2) << "During the months of " << month\_start << "-" << month\_end <<

", the total rainfall was " << total\_precip <<

" inches and the average monthly rainfall was " <<

(total\_precip / total\_rain) << " inches" << endl;

}



**Chapter 6**

6)

// Shipping Charges

// Sam Mallet 6/27/2021

#include <iostream>

#include <iomanip>

using namespace std;

double calculateCharge(double weight, int distance);

int main()

{

int distance;

double weight;

while (true)

{

cout << "Enter the weight of the package and the distance it is to be shipped" << endl << "Weight (kg): ";

cin >> weight;

if (weight <= 0)

break;

cout << "Distance (miles): ";

cin >> distance;

double shipping\_cost = calculateCharge(weight, distance);

cout << fixed << setprecision(2) << "The shipping cost for a " << weight << "kg package " << distance << " miles away is $" << shipping\_cost << endl << endl;

}

}

double calculateCharge(double weight, int distance)

{

double rate;

if (weight <= 2.0)

rate = 3.1;

else if (weight <= 6.0)

rate = 4.2;

else if (weight <= 10.0)

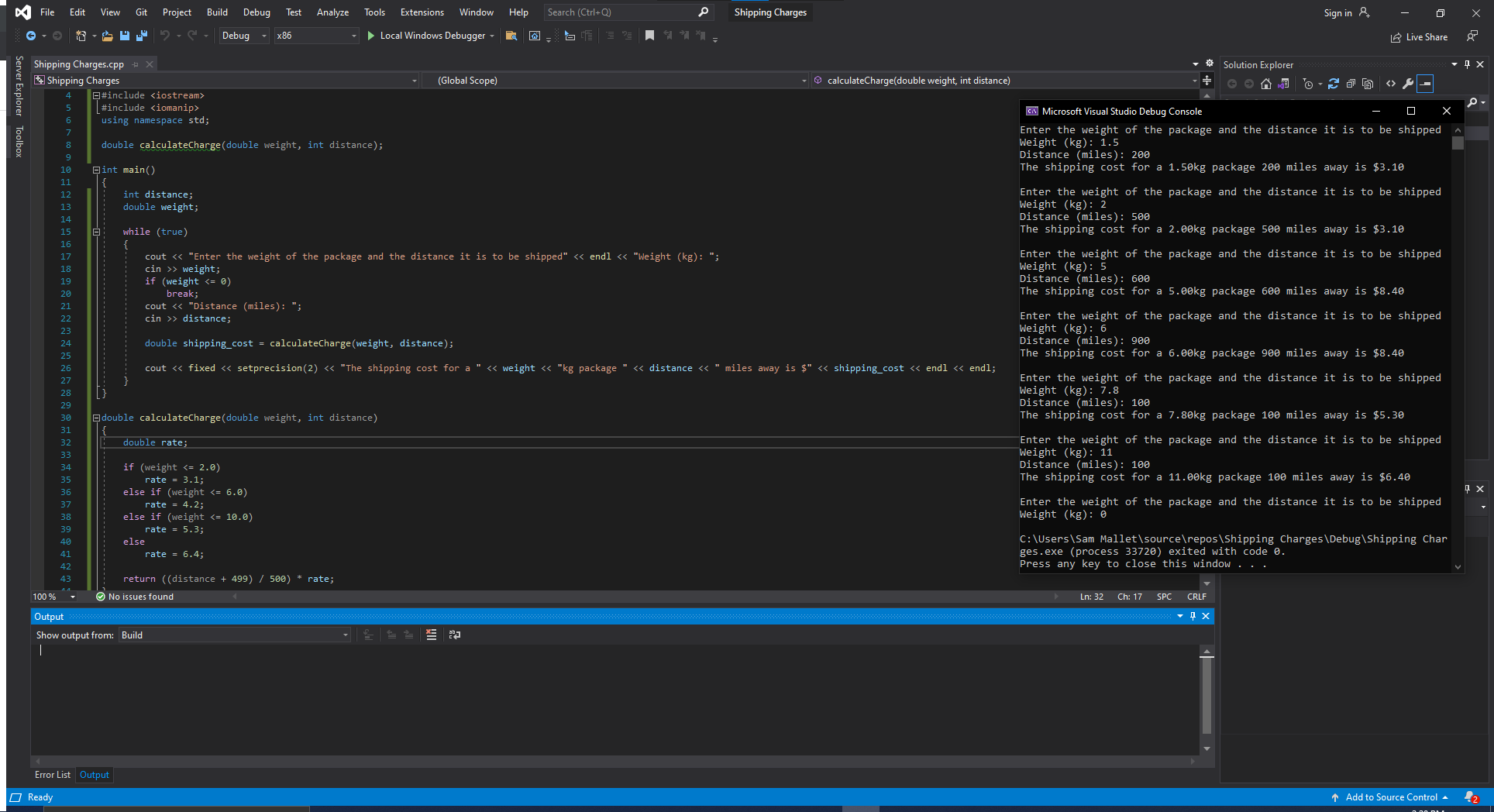
rate = 5.3;

else

rate = 6.4;

return ((distance + 499) / 500) \* rate;

}



9)

// Lowest Score Drop

// Sam Mallet 6/27/2021

#include <iostream>

using namespace std;

void getScore(int&);

void calcAverage(int, int, int, int, int, int);

int findLowest(int, int, int, int, int, int);

int main()

{

int s1, s2, s3, s4, s5, s6;

cout << "Enter you five test scores below to calculate the average after exempting your lowest:" << endl;

getScore(s1);

getScore(s2);

getScore(s3);

getScore(s4);

getScore(s5);

getScore(s6);

calcAverage(s1, s2, s3, s4, s5, s6);

}

void getScore(int& score)

{

cout << "Enter your score: ";

cin >> score;

while (score > 100 || score < 0)

{

cout << "Score must be between 0 and 100 (inclusive). Try again: ";

cin >> score;

}

}

void calcAverage(int s1, int s2, int s3, int s4, int s5, int s6)

{

double average = (s1 + s2 + s3 + s4 + s5 + s6 - findLowest(s1, s2, s3, s4, s5, s6)) / 5.0;

cout << "the average of your five scores minus the lowest is " << average;

}

int findLowest(int s1, int s2, int s3, int s4, int s5, int s6)

{

if (s1 < s2 && s1 < s3 && s1 < s4 && s1 < s5 && s1 < s6)

return s1;

else if (s2 < s1 && s2 < s3 && s2 < s4 && s2 < s5 && s2 < s6)

return s2;

else if (s3 < s2 && s3 < s1 && s3 < s4 && s3 < s5 && s3 < s6)

return s3;

else if (s4 < s2 && s4 < s3 && s4 < s1 && s4 < s5 && s4 < s6)

return s4;

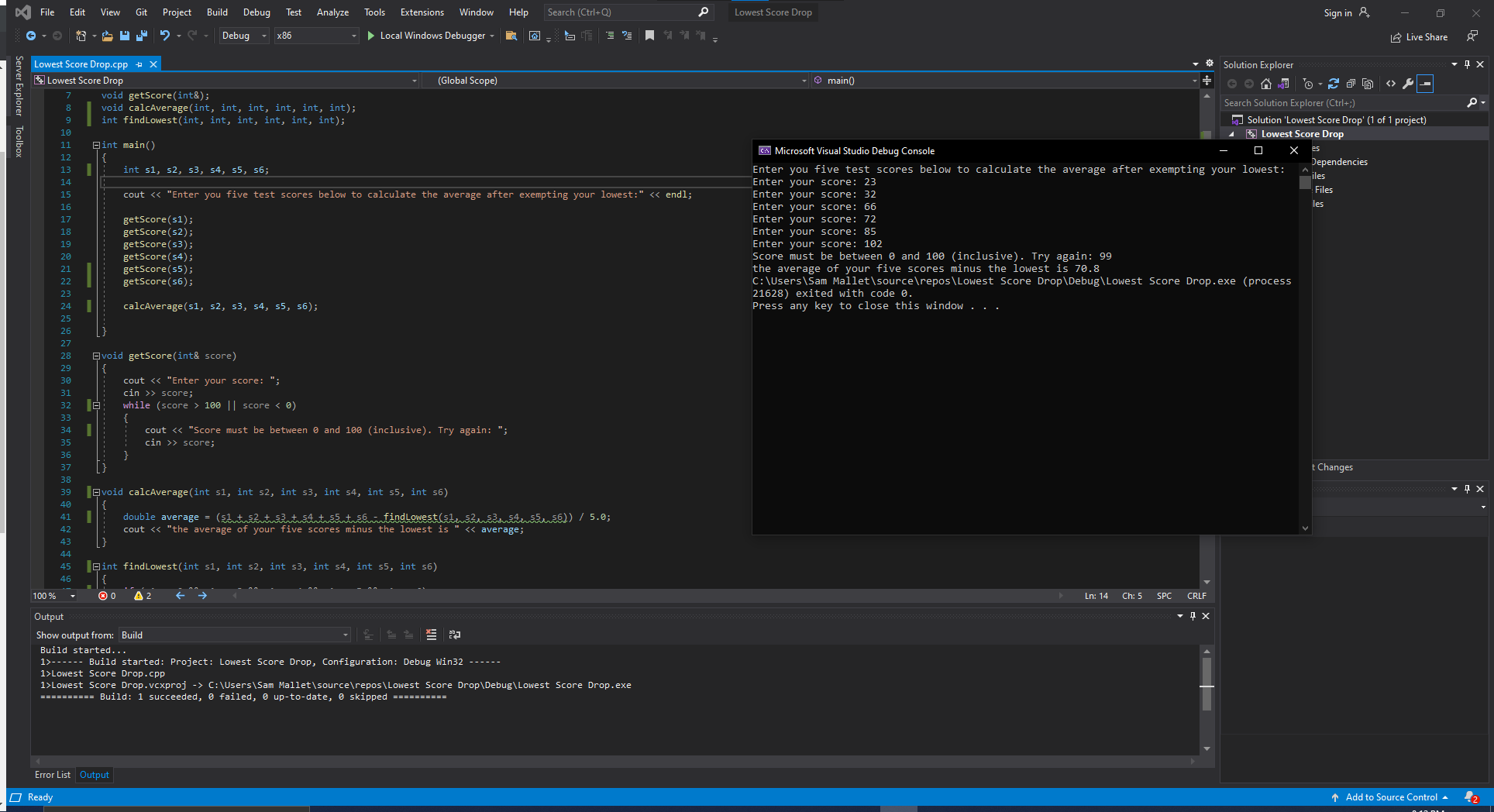
else if (s5 < s2 && s5 < s3 && s5 < s1 && s5 < s6 && s5 < s4)

return s5;

else

return s6;

}



15)

// Order Status

// Sam Mallet 6/27/2021

#include <iostream>

using namespace std;

void getInfo(int&, int&, double&);

void displayInfo(int, int, double);

int main(int&, int&, double&)

{

int ordered\_spools, spools\_in\_stock, again;

double ship\_charges;

do

{

getInfo(ordered\_spools, spools\_in\_stock, ship\_charges);

displayInfo(ordered\_spools, spools\_in\_stock, ship\_charges);

cout << endl << "Enter anything to contine and 0 to quit: ";

cin >> again;

} while (again != 0);

}

void getInfo(int& ordered, int& stock, double& charges)

{

cout << "Enter the number of spools ordered: ";

cin >> ordered;

cout << "Enter the number of spools in stock: ";

cin >> stock;

cout << "Enter any special shipping and handling charges above the regular $10 rate): ";

cin >> charges;

}

void displayInfo(int ordered, int stock, double charges = 10.0)

{

cout << "The number of ordered spools ready to ship from current stock: " << stock << endl;

if (ordered > stock)

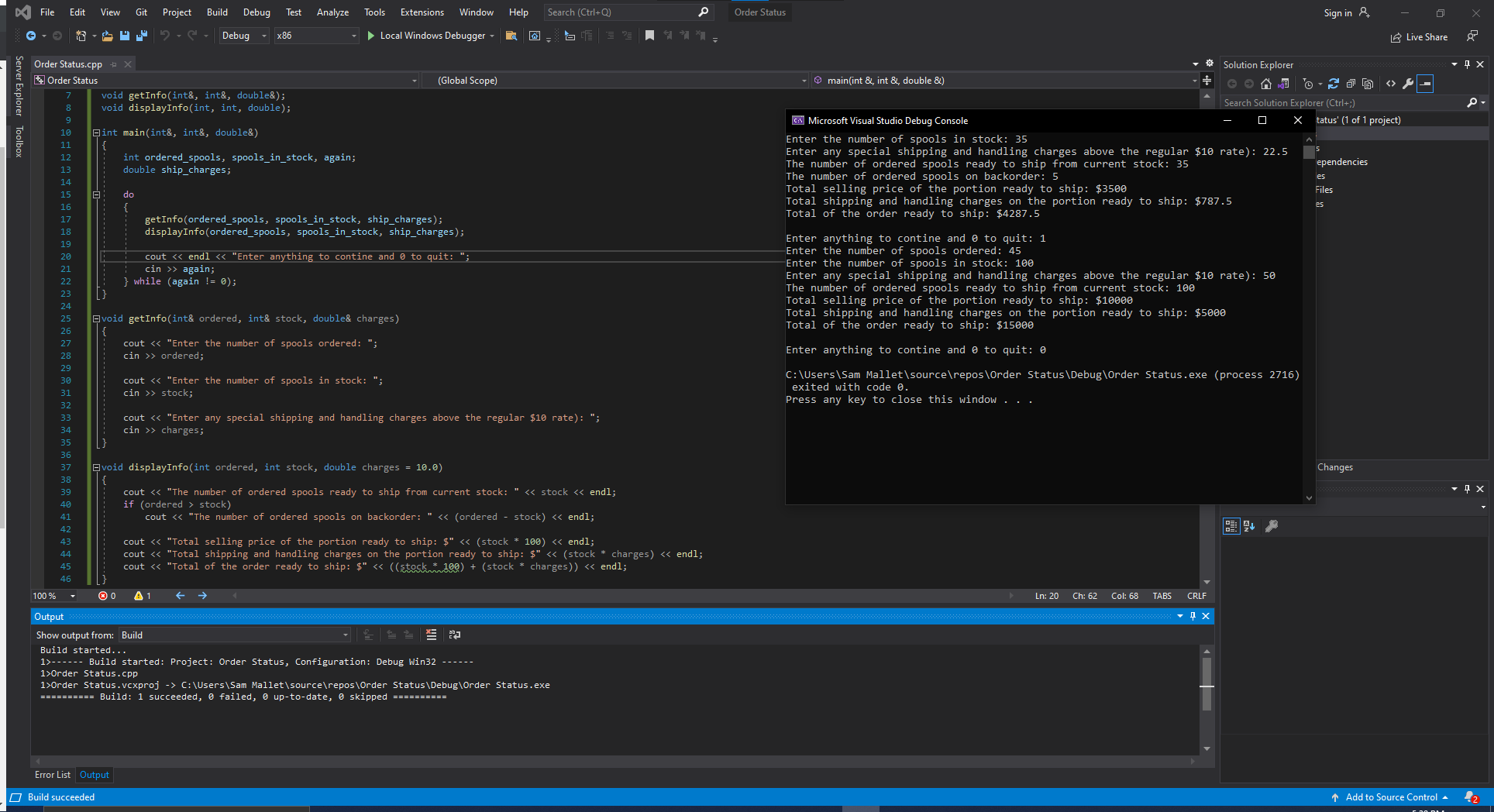
cout << "The number of ordered spools on backorder: " << (ordered - stock) << endl;

cout << "Total selling price of the portion ready to ship: $" << (stock \* 100) << endl;

cout << "Total shipping and handling charges on the portion ready to ship: $" << (stock \* charges) << endl;

cout << "Total of the order ready to ship: $" << ((stock \* 100) + (stock \* charges)) << endl;

}



16)

// Overloaded Hospital

// Sam Mallet 4/27/2021

#include <iostream>

#include <cctype>

using namespace std;

void checkInput(double&);

double calcCharges(double, double, double, double);

double calcCharges(double, double);

int main()

{

int again = 1;

while (again != 0)

{

int patient\_type;

double days, daily\_rate, service\_charges, med\_charges;

cout << "If you are an inpatient enter 0. If you are an outpatient enter 1: ";

cin >> patient\_type;

if (patient\_type == 0)

{

cout << "Enter the number of days spent in the hospital: ";

cin >> days;

checkInput(days);

cout << "Enter the daily rate: ";

cin >> daily\_rate;

checkInput(daily\_rate);

cout << "Enter the charges for hospital services (lab tests, etc.): ";

cin >> service\_charges;

checkInput(service\_charges);

cout << "Enter the hospital medication charges: ";

cin >> med\_charges;

checkInput(med\_charges);

cout << "The total charges from your hospital visit amounts to $" << calcCharges(days, daily\_rate, service\_charges, med\_charges) << endl << endl;

}

else if (patient\_type == 1)

{

cout << "Enter the charges for hospital services (lab tests, etc.): ";

cin >> service\_charges;

checkInput(service\_charges);

cout << "Enter the hospital medication charges: ";

cin >> med\_charges;

checkInput(med\_charges);

cout << "The total charges from your hospital visit amounts to $" << calcCharges(service\_charges, med\_charges) << endl << endl;

}

cout << "Enter 0 to quit or 1 to continue: ";

cin >> again;

}

}

void checkInput(double& input)

{

if (input < 0)

{

cout << "This value cannot be below 0: Please re-enter: ";

cin >> input;

}

}

double calcCharges(double days, double daily\_rate, double service\_charges, double med\_charges)

{

return (days \* daily\_rate) + service\_charges + med\_charges;

}

double calcCharges(double service\_charges, double med\_charges)

{

return service\_charges + med\_charges;

}

