Here is a link to the GitHub repository in case the code doesn’t work:

<https://github.com/nothinfoyou/CppApplication1>

1. Which of the following are valid C++ identifiers?

Answer: A, B, D, E, H, J

1. Which of the following is a reserved word in C++?

Answer: D, E

1. Given:

int num1, num2, newNum; double x, y;

Which of the following assignments are valid? If an assignment is not valid, state the reason.

1. num1 = 35;

Valid: num1 is an int, and 35 is an integer, so this is a valid assignment.

1. newNum = num1 – num2;

Valid: Both num1 and num2 are integers, and the result of subtracting two integers is also an integer, which can be assigned to newNum (an integer).

1. num1 = 5; num2 = 2 + num1; num1 = num2 / 3;

Valid: num1 = 5; is valid since num1 is an integer.

num2 = 2 + num1; is valid because adding two integers gives an integer.

num1 = num2 / 3; is valid since dividing two integers results in an integer, and the result is assigned to num1.

1. num1 \* num2 = newNum;

Invalid: In C++, the left-hand side of an assignment must be a variable, not an expression. num1 \* num2 is an expression and cannot be assigned a value.

1. x = 12 \* num1 - 15.3;

Valid: x is a double, and the result of 12 \* num1 (an integer) subtracted by 15.3 (a double) is a double, which can be assigned to x.

1. num1 \* 2 = newNum + num2;

Invalid: The left-hand side (num1 \* 2) is an expression and not a variable, so it cannot be assigned a value.

1. x / y = x \* y;

Invalid: The left-hand side (x / y) is an expression, not a variable, so it cannot be assigned a value.

This page is dedicated to Question 4 and 5:

#include <iostream>

#include <cmath>

#include <algorithm>

#include <iomanip>

#include <string>

using namespace std;

int main()

{

int num1;

cout << "Enter the number of the question you would like to see (4 or 5): ";

cin >> num1;

//Use switch statement to handle question

switch (num1) {

default:

cout << "Invalid";

break;

case 4:

// Variables to store values and results

double a, b, c;

double x, y, z, avg, max\_val, min\_val;

// Prompt user to enter values for a, b, and c

cout << "Enter the values for a, b, and c: " << endl;

cin >> a >> b >> c;

// Calculate x using the quadratic formula (Note: Formula is incorrect)

// Correct formula should be (-b ± sqrt(b^2 - 4ac)) / (2a)

x = (-b + sqrt((b \* b) + (4 \* a))) / (2 \* c);

// Calculate y as a polynomial expression

y = (pow(a, 4) - (b \* pow(c, 2)));

// Calculate z based on powers of a and b, divided by c

z = ((pow(a, 2)) \* (pow(b, 3))) / c;

// Calculate the average of x, y, and z

avg = (x + y + z) / 3;

// Find the maximum and minimum values among x, y, and z

max\_val = max({ x, y, z });

min\_val = min({ x, y, z });

// Output the calculated values

cout << "X: " << x << endl;

cout << "Y: " << y << endl;

cout << "Z: " << z << endl;

cout << "Average: " << avg << endl;

cout << "Max: " << max\_val << endl;

cout << "Min: " << min\_val << endl;

break;

case 5:

string customerID, name;

int nights;

double roomRate;

char customerType; // 'G' for Golden, 'R' for Regular

double totalCharge, discount;

// Set precision for currency output

cout << fixed << setprecision(2);

// Input customer details

cout << "Enter customer ID: ";

cin >> customerID;

cout << "Enter customer name: ";

cin.ignore(); // To clear newline character from the buffer

getline(cin, name);

cout << "Enter number of nights: ";

cin >> nights;

cout << "Enter room rate: ";

cin >> roomRate;

cout << "Enter customer type (G for Golden, R for Regular): ";

cin >> customerType;

// Calculate total charge based on customer type and nights stayed

if (customerType == 'G') {

if (nights <= 2) {

discount = 0.30; // 30% discount for golden customers staying 2 nights or less

}

else {

discount = 0.50; // 50% discount for golden customers staying more than 2 nights

}

}

else if (customerType == 'R') {

if (nights <= 2) {

discount = 0.10; // 10% discount for regular customers staying 2 nights or less

}

else {

discount = 0.30; // 30% discount for regular customers staying more than 2 nights

}

}

else {

cout << "Invalid customer type!" << endl;

return 1; // Exit the program with an error code

}

// Calculate total charge

double totalBeforeDiscount = nights \* roomRate;

totalCharge = totalBeforeDiscount \* (1 - discount);

// Display customer information and total charge

cout << "\nCustomer ID: " << customerID << endl;

cout << "Customer Name: " << name << endl;

cout << "Number of Nights: " << nights << endl;

cout << "Room Rate: $" << roomRate << endl;

cout << "Discount Applied: " << (discount \* 100) << "%" << endl;

cout << "Total Charge: $" << totalCharge << endl;

break;

}

}