**Daily Assessment**

**Muhammad Ammar Chaudhry**

**SU92-BSCSM-F23-269**

**BSCS-1G**

**----------------------------------------------------------------------------------------------------**

**Q # 81:**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int userInput;

double squareRoot;

cout << "Enter a positive integer: ";

cin >> userInput;

if (userInput < 0)

{

cout << "Error: Please enter a positive integer." << endl;

return 0;

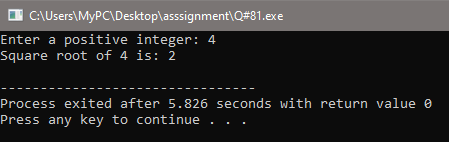
}

squareRoot = sqrt(userInput);

cout << "Square root of " << userInput << " is: " << squareRoot << endl;

return 0;

}



**Q # 82:**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double radius, area;

cout << "Enter the radius of the circle: ";

cin >> radius;

if (radius < 0)

{

cout << "Error: Please enter a non-negative radius." << endl;

return 1;

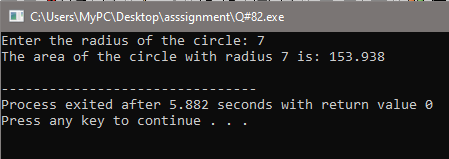
}

area = M\_PI \* pow(radius, 2);

cout << "The area of the circle with radius " << radius << " is: " << area << endl;

return 0;

}



**Q # 83:**

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

const int MAX\_LENGTH = 100;

char string1[MAX\_LENGTH], string2[MAX\_LENGTH];

cout << "Enter the first string: ";

cin.getline(string1, MAX\_LENGTH);

cout << "Enter the second string: ";

cin.getline(string2, MAX\_LENGTH);

char concatenatedString[MAX\_LENGTH \* 2];

strcpy(concatenatedString, string1);

strcat(concatenatedString, string2);

int lengthString1 = strlen(string1);

int lengthString2 = strlen(string2);

int comparisonResult = strcmp(string1, string2);

cout << "Concatenated string: " << concatenatedString << endl;

cout << "Length of the first string: " << lengthString1 << endl;

cout << "Length of the second string: " << lengthString2 << endl;

if (comparisonResult == 0)

{

cout << "The strings are equal." << endl;

}

else

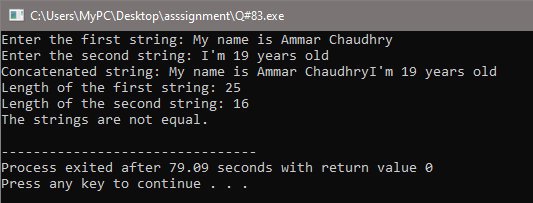
{

cout << "The strings are not equal." << endl;

}

return 0;

}



**Q # 84:**

#include <iostream>

using namespace std;

float calculateTotal(float price, int quantity)

{

float total = price \* quantity;

return total;

}

int main()

{

float price = 5.99;

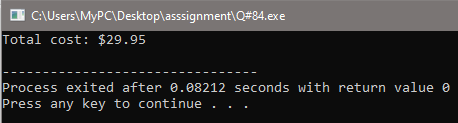
int quantity = 5;

float totalCost = calculateTotal(price, quantity);

cout << "Total cost: $" << totalCost << endl;

return 0;

}



**Q # 85:**

#include <iostream>

using namespace std;

void swapNumbers(int& a, int& b)

{

int temp = a;

a = b;

b = temp;

}

int main()

{

int num1, num2;

cout << "Enter the first integer: ";

cin >> num1;

cout << "Enter the second integer: ";

cin >> num2;

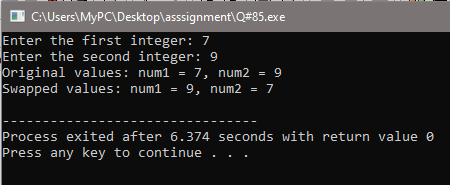
cout << "Original values: num1 = " << num1 << ", num2 = " << num2 << endl;

swapNumbers(num1, num2);

cout << "Swapped values: num1 = " << num1 << ", num2 = " << num2 << endl;

return 0;

}



**Q # 86:**

#include <iostream>

using namespace std;

float calculateArea(float length, float width)

{

return length \* width;

}

float calculateArea(float radius)

{

return 3.14159 \* radius \* radius;

}

int main()

{

float rectangleLength = 10;

float rectangleWidth = 5;

float circleRadius = 7;

float rectangleArea = calculateArea(rectangleLength, rectangleWidth);

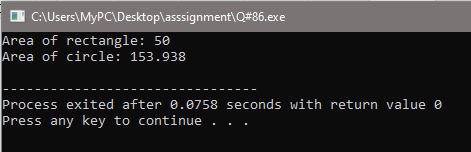
float circleArea = calculateArea(circleRadius);

cout << "Area of rectangle: " << rectangleArea << endl;

cout << "Area of circle: " << circleArea << endl;

return 0;

}



**Q # 87:**

#include <iostream>

using namespace std;

float calculateCost(float pricePerUnit, int quantity, float discountPercentage = 0.0)

{

float totalCost = pricePerUnit \* quantity \* (1.0 - discountPercentage / 100.0);

return totalCost;

}

int main()

{

float price = 10.0;

int quantity = 5;

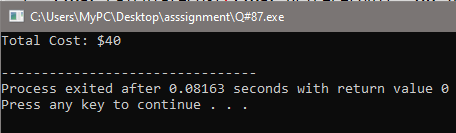
float discount = 20.0;

float cost = calculateCost(price, quantity, discount);

cout << "Total Cost: $" << cost << endl;

return 0;

}



**Q # 88:**

#include <iostream>

using namespace std;

int calculateFactorial(int n)

{

if (n == 0)

{

return 1;

}

return n \* calculateFactorial(n - 1);

}

int main()

{

int userInput;

cout << "Enter a positive integer: ";

cin >> userInput;

if (userInput < 0)

{

cerr << "Error: Please enter a non-negative integer." << endl;

return 1;

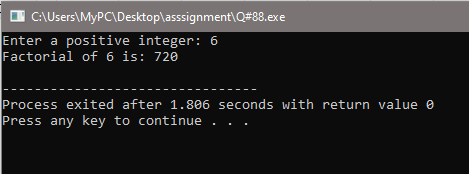
}

int result = calculateFactorial(userInput);

cout << "Factorial of " << userInput << " is: " << result << endl;

return 0;

}



**Q # 89:**

#include<iostream>

using namespace std;

inline float add(float a, float b)

{

return a + b;

}

inline float subtract(float a, float b)

{

return a - b;

}

inline float multiply(float a, float b)

{

return a \* b;

}

inline float divide(float a, float b)

{

return (b != 0) ? (a / b) : 0;

}

int main()

{

float num1, num2;

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

float result\_add = add(num1, num2);

float result\_subtract = subtract(num1, num2);

float result\_multiply = multiply(num1, num2);

float result\_divide = divide(num1, num2);

cout << "Addition: " << result\_add << endl;

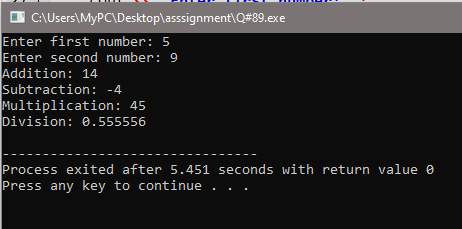
cout << "Subtraction: " << result\_subtract << endl;

cout << "Multiplication: " << result\_multiply << endl;

cout << "Division: " << result\_divide << endl;

return 0;

}



**Q # 90:**

#include <iostream>

using namespace std;

int globalVariable = 10;

void demonstrateScopes()

{

int globalVariable = 5;

cout << "Local variable within function: " << globalVariable << endl; // Used cout without std::

}

int main()

{

cout << "Global variable in main function: " << globalVariable << endl;

demonstrateScopes();

cout << "Global variable in main function after function call: " << globalVariable << endl;

return 0;

}

