***UFCFGL-30-1-Programming in C++***

ASSIGNMENT OF C++

University ID: 22085543

**Email Address:** [Krish2.Karki@live.uwe.ac.uk](mailto:Krish2.Karki@live.uwe.ac.uk)

Name : Krish Karki

College ID:9522

# TASKS

Task-1: [Total Marks- 20]

1. Explain what is wrong with the comments below and correct them. [2 marks]
2. /\* *this is a comment?* /\*

**/\* There is an extra '/' at the end of the comment, which causes it to not be properly closed. The corrected version would be:**

**/\* this is a comment? \*/**

1. /\* How about this one /\* it seems like a comment/\* doesn’t it? /

**/ There are two opening comment tags, but only one closing comment tag. This causes the comment to not be properly closed. The corrected version would be:**

**/\* How about this one? It seems like a comment, doesn't it? \*/**

1. Indicate which of the symbols below are C++ reserved words, or valid or invalid identifiers. [3 marks]

**float: valid identifier**

**cout: reserved word**

**Bill: valid identifier**

**"hello": Invalid identifier**

**rate: valid identifier**

**start: valid identifier**

**var: reserved word**

**xyz123: valid identifier**

**123xyz: valid identifier**

**'a': invalid identifier**

**include: reserved word**

**return: reserved word**

**5TBC: invalid identifier**

**TBC5: invalid identifier**

**TBC@1234: invalid identifier**

**TIME: reserved word**

1. What will be the output of the following code snippet? Choose the correct answer. [2 marks]

bool x = 20;

cout << x ;

(a) 0

(b) false

(c) 1

(d) true

**(d) true**

1. Solve the following expression to get the final output? [3 marks]

i = 1;

i = (i<<= 1 % 2) ;

cout << i << endl ;

**OUTPUT:2**

1. For the following statements find the values generated for p and q? Your answer must contain the values of p and q in every step. [5 marks]

int p = 0, q =1;

p = q++;

p = ++q;

p = q – –;

p = – –q;

The value of p and q are \_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_ .

**Here is how the values of p and q change throughout the code:**

**1.int p = 0, q =1;**

**p is initialized to 0**

**q is initialized to 1**

**2.p = q++;**

**p is assigned the value of q, which is 1**

**q is then incremented to 2**

**3.p = ++q;**

**q is incremented to 3**

**p is assigned the value of q, which is 3**

**4.p = q --;**

**p is assigned the value of q, which is 3**

**q is then decremented to 2**

**5.p = --q;**

**q is decremented to 1**

**p is assigned the value of q, which is 1**

**Therefore, the final values of p and q are 1 and 1, respectively.**

1. Solve the following expression[step by step]to get the final result of x.[5 marks]

int x = 3<<4+5<<6;

**Here is how the expression can be evaluated step by step:**

**1.int x = 3<<4+5<<6;**

**First, the expression 4+5 is evaluated, resulting in 9.**

**2.int x = 3<<9<<6;**

**Then, the expression 3<<9 is evaluated, resulting in 768.**

**3.int x = 768<<6;**

**Finally, the expression 768<<6 is evaluated, resulting in 49152.**

**Therefore, the final result of x is 49152.**

Task-2: Debugging\_task\_1**[Total - 30 Marks]**

1. Show the output displayed by the program lines below when the data entered are 3.0 and 5.0. [5 Marks]

cout << "Enter two numbers: ";

cin >> a >> b;

a = a - 5.0;

b = a \* b;

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

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

**The following output would be displayed:**

Enter two numbers:

a = -2

b = -10

**This is because the value of a is first set to a - 5.0, which is 3.0 - 5.0 or -2. Then, b is set to a \* b, which is (-2) \* 5.0 or -10.**

2. What value is assigned to x for each segment below when y is 10.0?

**a.** x = 20.0;

if (y != (x - 10.0))

x = x - 10.0;

else

x = x / 2.0;

b. if (y &lt; 5.0 &amp;&amp; y &gt;= 0.0)

x = 5 + y;

else

x = 2 - y;

a. In this segment, y is 10.0 and x is 20.0. The condition in the if statement is y != (x - 10.0),

which is equivalent to y != 10.0. Since y is not equal to 10.0, the condition is false and the

value of x is not changed. Therefore, the value of x after this segment is 20.0.

1. For segment b, if y is less than 5.0 and greater than or equal to 0.0, then x is assigned the value of 5 + y, which is 15.0. If y is not in this range, x is assigned the value of 2 - y, which is -8.0.

3Insert braces where needed below to avoid syntax or logic errors. Indent as needed to improve readability. The last statement should execute regardless of the value of x or y. [10 Marks]

if (x > y)

x = x + 10.0;

cout << "x bigger than y" << endl;

else

y = y + 10.0;

cout << "x smaller than y" << endl;

cout << "x is " << x

<< "y is " << y << endl;

if (x > y) {

x = x + 10.0;

cout << "x bigger than y" << endl;

}

else {

y = y + 10.0;

cout << "x smaller than y" << endl;

}

cout << "x is " << x << " y is " << y << endl;

4. What does the while statement below display? Rewrite it as a for statement and as a do-while statement. [10 Marks]

num = 5;

while (num <= 50)

{

cout << num << endl;

num += 5;

}

The while statement displays the numbers 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 (each on a new line).

Here is the equivalent for loop:

for (int num = 5; num <= 50; num += 5) {

cout << num << endl;

}

Here is the equivalent do-while loop:

num = 5;

do {

cout << num << endl;

num += 5;

} while (num <= 50);

Task-3: [Total Marks-50]

1.Write the declarations for a program that converts a weight in pounds to a weight in kilograms.

#include <iostream>

using namespace std;

double pounds;

double kilograms;

const double POUNDS\_PER\_KILOGRAM = 2.20462;

int main() {

// program code goes here

return 0;

}

1. Write a program that reads two int values into *m*, *n* and displays their sum, their differences (*m* – *n* and *n* – *m*), their product, their quotients (*m* / *n* and *n* / *m*) and both *m* % *n* and *n* % *m*. If the numbers are 4 and 5, the line that shows their sum should be displayed as: 5 + 4 = 9

#include <iostream>

int main() {

int m, n;

std::cin >> m >> n;

// Calculate and display sum

int sum = m + n;

std::cout << m << " + " << n << " = " << sum << std::endl;

// Calculate and display differences

int diff1 = m - n;

int diff2 = n - m;

std::cout << m << " - " << n << " = " << diff1 << std::endl;

std::cout << n << " - " << m << " = " << diff2 << std::endl;

// Calculate and display product

int product = m \* n;

std::cout << m << " \* " << n << " = " << product << std::endl;

// Calculate and display quotients

int quotient1 = m / n;

int quotient2 = n / m;

std::cout << m << " / " << n << " = " << quotient1 << std::endl;

std::cout << n << " / " << m << " = " << quotient2 << std::endl;

// Calculate and display remainders

int remainder1 = m % n;

int remainder2 = n % m;

std::cout << m << " % " << n << " = " << remainder1 << std::endl;

std::cout << n << " % " << m << " = " << remainder2 << std::endl;

return 0;

}

3. Implement the decision table below using a nested if statement. Assume that the grade point average is within the range 0.0 through 4.0.

**Grade Point Average** **Transcript Message**

0.0 to 0.99 Failed semester—registration suspended

1.0 to 1.99 On probation for next semester

2.0 to 2.99 (no message)

3.0 to 3.49 Dean’s list for semester

3.5 to 4.0 Highest honors for semester

#include <iostream>

int main() {

// Read in grade point average

double gpa;

std::cout << "Enter your grade point average: ";

std::cin >> gpa;

// Determine transcript message

if (gpa >= 0.0 && gpa <= 0.99) {

std::cout << "Failed semester - registration suspended" << std::endl;

} else if (gpa >= 1.0 && gpa <= 1.99) {

std::cout << "On probation for next semester" << std::endl;

} else if (gpa >= 2.0 && gpa <= 2.99) {

// No message for this range

} else if (gpa >= 3.0 && gpa <= 3.49) {

std::cout << "Dean's list for semester" << std::endl;

} else if (gpa >= 3.5 && gpa <= 4.0) {

std::cout << "Highest honors for semester" << std::endl;

} else {

std::cout << "Invalid grade point average" << std::endl;

}

return 0;

}

1. There are 900 people in a town whose population increases by 10 percent (a data item) each year. Write a loop that displays the annual population and determines how many years (countYears) it will take for the population to pass 20,000. Verify that your program works if the population doubles each year (an increase of 100%).

#include <iostream>

int main() {

int population = 900;

int countYears = 0;

double increase = 0.1; // 10% increase per year

while (population <= 20000) {

population += population \* increase;

countYears++;

std::cout << "Year " << countYears << ": " << population << std::endl;

}

std::cout << "It will take " << countYears << " years for the population to pass 20,000." << std::endl;

return 0;

}

This program uses a while loop to increment the population by 10% each year until it exceeds 20,000. It also keeps track of the number of years that have passed using the countYears variable.

To modify the program to double the population each year (an increase of 100%), you can simply change the increase variable to 1.0:

int population = 900;

int countYears = 0;

double increase = 1.0; // 100% increase per year

while (population <= 20000) {

population += population \* increase;

countYears++;

std::cout << "Year " << countYears << ": " << population << std::endl;

}

std::cout << "It will take " << countYears << " years for the population to pass 20,000." << std::endl;

## 