Jeffrey Cho

Chapter 5 Study Guide

1. What is the output of the following C++ code?

int i = 2; int temp = 1;

while (i < 10)

{

temp = temp \* (i - 1); i = i + 1;

}

cout << "i = " << i << " and temp = " << temp << endl;

**i = 10 and temp = 40320**

1. Suppose that the input is 1 20 35 15 28 66 -1. What is the output of the following code? Is this a count-controlled loop or a sentinel-controlled loop?

int num; int sum; int count; cin >> count; cin >> sum;

while (count <= 4)

{

cin >> num; sum = sum + num; count++;

}

cout << "Sum = " << sum << endl;

**sum = 164**

**This is a count-controlled loop**

1. Suppose that the input is 3 4 6 7 2 -1. What is the output of the following code? Is this a count-controlled loop or a sentinel-controlled loop?

int num; int sum; cin >> sum; num = sum;

while (num != -1)

{

cin >> num; sum = sum + 2 \* num; }

cout << "Sum = " << sum << endl;

**sum = 39**

**This is a sentinel - controlled loop**

1. When does the following while loop terminate?

ch = 'D';

while ('A' <= ch && ch <= 'Z') ch = static\_cast<char>(static\_cast<int>(ch) + 1);

**The loop terminates after ch runs through and becomes ‘Z’. When ch becomes ‘[‘, the loop terminates.**

1. The following program is designed to input two numbers and output their sum. It asks the user if he/she would like to run the program. If the answer is Y or y, it prompts the user to enter two numbers. After adding the numbers and displaying the results, it again asks the user if he/she would like to add more numbers. However, the program fails to do so. Correct the program so that it works properly.

#include <iostream> #include <iomanip> using namespace std; int main()

{

char response; double num1; double num2;

cout << "This program adds two numbers." << endl; cout << "Would you like to run the program: (Y/y) "; cin >> response; cout << endl;

cout << fixed << showpoint << setprecision(2); while (response == 'Y' ***||*** response == 'y')

{

cout << "Enter two numbers: "; cin >> num1 >> num2; cout << endl;

cout << num1 << " + " << num2 << " = " << (num1 ***+*** num2) << endl;

cout << "Would you like to add again: (Y/y) "; cin >> response; cout << endl;

} return 0;

}

1. What is the output of the following program segment?

int num = 1; int i;

for (i = 0; i < 5; i++)

{

num = num \* (5 - i);

cout << num << " ";

}

cout << endl;

**5 20 60 120 120**

1. Assume that the following code is correctly inserted into a program:

int s = 0; int i;

for (i = 0; i < 5; i++)

{

s = 2 \* s + i;

cout << s << " ";

}

cout << endl;

* 1. What is the final value of s? (i) 11 (ii) 4 ***(iii) 26*** (iv) none of these
  2. If a semicolon is inserted after the right parenthesis in the for loop statement, what is the final value of s?

(i) 0 (ii) 1 (iii) 2 ***(iv) 5*** (v) none of these

* 1. If the 5 is replaced with a 0 in the for loop control expression, what is the final value of s? ***(i) 0*** (ii) 1 (iii) 2 (iv) none of these

1. What is the output of the following code? Is there a relationship between the variables x and y? If yes, state the relationship? What is the output?

int x = 19683; int i; int y = 0;

for (i = x; i >= 1; i = i / 3)

y++;

cout << "x = " << x << ", y = " << y << endl;

**There is a relationship between variables x and y. The relationship between x and y is that x is 3y.**

**Output: x = 19683 y = 10**

1. Use a do..while loop to write an input statement validation loop that prompts the user to enter a number less than 20 or greater than 75.

**int num1;**

**do**

**{**

**cout << "Please enter a number less than 20 or greater than 75: ";**

**cin >> num1;**

**}**

**while (20 <= num1 && num1 <= 75);**

1. Which of the following apply to the while loop only? To the do...while loop only? To both?
   1. It is considered a conditional loop. **both**
   2. The body of the loop executes at least once. **do…while**
   3. The logical expression controlling the loop is evaluated before the loop is entered. **while**
   4. The body of the loop may not execute at all. **while**

1. Rewrite the following as a for loop:

int i = 0, value = 0;

while (i <= 20)

{

if (i % 2 == 0 && i <= 10) value = value + i \* i; else if (i % 2 == 0 && i > 10) value = value + i; else

value = value - i; i = i + 1;

}

cout << "value = " << value << endl;

**for loop**

**int i = 0, value = 0;**

**for (i; i<=20; i++){**

**if (i % 2 == 0 && i <= 10) value = value + i \* i;**

**else if (i % 2 == 0 && i > 10) value = value + i;**

**else**

**value = value - i; }**

**cout << "value = " << value << endl;**

What is the output of this loop?

**value = 200**

Rewrite the while loop as a do...while loop

**int i = 0, value = 0;**

**do{**

**if (i % 2 == 0 && i <= 10) value = value + i \* i; else if (i % 2 == 0 && i > 10) value = value + i; else**

**value = value - i; i = i + 1;**

**}**

**while (i <= 20) ;**

**cout << "value = " << value << endl;**

1. To learn how nested for loops work, do a walk-through of the following program segments and determine, in each case, the exact output.
   1. int i, j;

for (i = 1; i <= 5; i++)

{

for (j = 1; j <= 5; j++) cout << setw(3) << i;

cout << endl;

}

**1 1 1 1 1**

**2 2 2 2 2**

**3 3 3 3 3**

**4 4 4 4 4**

**5 5 5 5 5**

* 1. int i, j;

for (i = 1; i <= 5; i++)

{

for (j = (i + 1); j <= 5; j++) cout << setw(5) << j;

cout << endl;

}

**2 3 4 5**

**3 4 5**

**4 5**

**5**

* 1. int i, j;

for (i = 1; i <= 5; i++)

{

for (j = 1; j <= i; j++) cout << setw(3) << j;

cout << endl;

}

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

* 1. const int M = 10; const int N = 10; int i, j;

for (i = 1; i <= M; i++)

{

for (j = 1; j <= N; j++) cout << setw(3) << M \* (i - 1) + j; cout << endl;

}

**\_\_1\_\_2\_\_3\_\_4\_\_5\_\_6\_\_7\_\_8\_\_9\_10**

**\_11\_12\_13\_14\_15\_16\_17\_18\_19\_20**

**\_21\_22\_23\_24\_25\_26\_27\_28\_29\_30**

**\_31\_32\_33\_34\_35\_36\_37\_38\_39\_40**

**\_41\_42\_43\_44\_45\_46\_47\_48\_49\_50**

**\_51\_52\_53\_54\_55\_56\_57\_58\_59\_60**

**\_61\_62\_63\_64\_65\_66\_67\_68\_69\_70**

**\_71\_72\_73\_74\_75\_76\_77\_78\_79\_80**

**\_81\_82\_83\_84\_85\_86\_87\_88\_89\_90**

**\_91\_92\_93\_94\_95\_96\_97\_98\_99100**

**\*\* \_ are blank spaces**

* 1. int i, j;

for (i = 1; i <= 9; i++)

{

for (j = 1; j <= (9 - i); j++) cout << " "; for (j = 1; j <= i; j++) cout << setw(1) << j; for (j = (i - 1); j >= 1; j--) cout << setw(1) << j;

cout << endl;

}

**\_\_\_\_\_\_\_\_1**

**\_\_\_\_\_\_\_121**

**\_\_\_\_\_\_12321**

**\_\_\_\_\_1234321**

**\_\_\_\_123454321**

**\_\_\_12345654321**

**\_\_1234567654321**

**\_123456787654321**

**12345678987654321**

**\*\* \_ are blank spaces**

1. What is the output of the following code?

int num = 12;

while (num >= 0)

{

if (num % 5 == 0) break; cout << num << " ";

num = num - 2;

}

cout << endl;

**12**

1. What is the output of the following code?

int num = 12;

while (num >= 0)

{

if (num % 5 == 0)

{

num++;

continue;

}

cout << num << " ";

num = num - 2;

}

cout << endl;

**12 11 9 7 6 4 2 1**