# Preincrement and Postincrement

Assume that total, num1, and num2 are int variables, with total = 2,

num1 = 3, and num2 = 4. Show the new values of all variables changed

by each of the following:

1. total = ++num1;

**total = 4;**

**num1 = 4;**

1. total = num2++;

**total = 4;**

**num2 = 5;**

1. total \*= 5;

**total = 10;**

1. total += num1 - num2 \* 4;

**total = -11;**

1. total = ++num1 + num2++;

**total = 8;**

**num1 = 4;**

**num2 = 5;**

1. total += 3 \* num1++ - 5;

**total = 6;**

**num1 = 4;**

# Code Trace

Show the output produced by the following segments of code. Assume that all variables are declared as int.

7. i = 30; 8. i = 10;

j = 10; j = 10;

if (i > 10 + j) if (i > 10 + j)

while (i > j) while (i > j)

{ {

i = i - 10; i = i - 10;

cout << i << endl; cout << i << endl;

} }

else else

while (j >= i) while (j >= i)

{ {

j = j - 10; j = j - 10;

cout << j << endl; cout << j << endl;

} }

9. i = 30; 10. for (i = 1 ; i <= 10 ; i++)

j = 10; {

while (i >= j) if (i == 2)

{ cout << "hello" << endl;

if (i > j + 10) else if (i == 3)

cout << "yes\n"; cout << "goodbye" << endl;

else else if (i == 4)

cout << "no\n"; cout << "why" << endl;

i = i - 10; if (i < 5)

} cout << "help" << endl;

if (i > 6)

cout << "blue" << endl;

}

11. w = 0; 12. w = 0;

for (h = -2; h <= 5; ++h) for (h = 10; h > 0; --h)

w = w + h; cout << w;

cout << w;

13. n = 1; 14. for (x = 1; x <= 5; x++)

for (k = 2; k <= 5; k++) cout << x << endl;

{ cout << x << endl;

n = k - 2 \* 3;

cout << k << ' ' << n << endl;

}

**Output for 7:**

**20**

**10**

**Output for 8:**

**0**

**Output for 9:**

**yes**

**no**

**Output for 10:**

**help**

**hello**

**help**

**goodbye**

**help**

**why**

**help**

**blue**

**blue**

**blue**

**blue**

**Output for 11:**

**12**

**Output for 12:**

**0000000000**

**Output for 13:**

**2 -4**

**3 -3**

**4 -2**

**5 -1**

**Output for 14:**

**1**

**2**

**3**

**4**

**5**

**6**

## Code Trace with Loops and Sentinels

15. Given the following program segment:

sum = finished = count = 0;

while (count <= 8 && !finished)

{

cin >> number;

if (number > 0)

sum += number;

else if (number == 0)

finished = 1;

count++;

}

cout << sum << ' ' << count << endl;

What would be the output given this data?

9 -3 4 6 0 5 6 0 5

**Output:**

**19 5**

16. What does the following program segment do? (Your answer should just be a one or two sentence summary of what the code does.)

punctuation = letters = digits = 0;

cout << "enter any character or '/' to quit: ";

cin >> character;

while (character != '/')

{

if (character == ',' || character == '.')

punctuation++;

else if (character >= 'a' && character <= 'z' ||

character >= 'A' && character <= 'Z')

letters++;

else if (character >= '0' && character <= '9')

digits++;

cout << "enter any character or '/' to quit: ";

cin >> character;

}

cout << punctuation << ' ' << letters << ' ' << digits << endl;

**The program takes in characters one at time from the user until they quit with ‘/’ and shows the number of characters that were punctuation, letters, and digits.**

# Programming Exercises

17. Input a character and a number from the user. Print "number" number of lines of output with the character printed once on each line.

#include <iostream>

using namespace std;

int main()

{

char in;

int number;

cout << "Enter a character and number: ";

cin >> in >> number;

for (int i = 0; i < number; ++i) {

cout << in << '\n';

}

}

18. Write a program segment (including a while loop) that reads one number into a variable named n and reads another number into a variable called maxPower. Then your statements should raise n to each power 0, 1, 2, ..., maxPower, and print a table like the following (suppose n is 2 and maxPower is 4):

n raised

n power to power

--------------------------

2 0 1

2 1 2

2 2 4

2 3 8

2 4 16

int n;

int maxPower;

int currentPower = 0;

cin >> n >> maxPower;

cout << "\t\t"

<< "n raised"

<< "\n";

cout << "n"

<< "\t"

<< "power"

<< "\t"

<< "to power"

<< "\n";

cout << "--------------------------"

<< "\n";

while (currentPower <= maxPower) {

cout << n << "\t"

<< currentPower << "\t"

<< pow(n, currentPower)

<< "\n";

currentPower++;

}

19. Redo problem 18 using a “for” loop.

int n;

int maxPower;

cin >> n >> maxPower;

cout << "\t\t"

<< "n raised"

<< "\n";

cout << "n"

<< "\t"

<< "power"

<< "\t"

<< "to power"

<< "\n";

cout << "--------------------------"

<< "\n";

for (int currentPower = 0; currentPower <= maxPower; ++currentPower) {

cout << n << "\t"

<< currentPower << "\t"

<< pow(n, currentPower)

<< "\n";

}

20. Given the following code, change the loop to an event controlled

loop that will exit when the user enters a negative number of hours

as the sentinel (Meaning the user will no longer need to enter the number of employees). **Do not use the break or exit command!!!**:

cout << "Enter the number of employees: ";

cin >> numEmployees;

totpay = 0;

empCount = 0;

while (empCount < numEmployees)

{

cout << "Hours: ";

cin >> hours;

cout << "Rate: $";

cin >> rate;

pay = hours \* rate;

cout << "Employee pay is : $ " << pay << endl;

totpay += pay;

empCount++;

}

cout << "Total payroll is $ " << totpay << endl;

cout << "Enter the number of employees: ";

cin >> numEmployees;

totpay = 0;

empCount = 0;

bool done = false;

while (empCount < numEmployees && !done) {

cout << "Hours: ";

cin >> hours;

if (hours < 0 && !done) {

done = true;

} else {

cout << "Rate: $";

cin >> rate;

pay = hours \* rate;

cout << "Employee pay is : $ " << pay << endl;

totpay += pay;

empCount++;

}

}

cout << "Total payroll is $ " << totpay << endl;

21. Write the statements to read in a group of exam scores ranging in

value from 0 to 100. Your program should count and print the number

of outstanding scores (90 to 100), satisfactory scores (70 to 89), and

the number of unsatisfactory scores (0 to 69). Stop reading exam scores when a negative value is entered.

int current\_score = 0;

int outstanding\_count = 0;

int satisfactory\_count = 0;

int unsatisfactory\_count = 0;

while (current\_score >= 0) {

cin >> current\_score;

if (current\_score >= 90 && current\_score <= 100) {

outstanding\_count++;

} else if (current\_score >= 70 && current\_score <= 89) {

satisfactory\_count++;

} else if (current\_score >= 0 && current\_score <= 69) {

unsatisfactory\_count++;

}

}

cout << "Outstanding Scores: " << outstanding\_count << "\n";

cout << "Satisfactory Scores: " << satisfactory\_count << "\n";

cout << "Unsatisfactory Scores: " << unsatisfactory\_count << "\n";