



Students:

Section 1.4 is a part of 1 assignment: **CSC108 CH01 PA**

Includes: PA



This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See [this article](#) for more info.

Due: 01/30/2025, 11:59 PM EST

1.4 Comments and whitespace

Comments

A **comment** is text a programmer adds to code, to be read by humans to better understand the code but ignored by the compiler. Two common kinds of comments exist:

- A **single-line comment** starts with `//` and includes all the following text on that line. Single-line comments commonly appear after a statement on the same line.
- A **multi-line comment** starts with `/*` and ends with `*/`, where all text between `/*` and `*/` is part of the comment. A multi-line comment is also known as a **block comment**

Figure 1.4.1: Comments example.

```
#include <iostream>
using namespace std;

/*
 * This program calculates the amount of pasta to cook, given the
 * number of people eating.
 *
 * Author: Andres Sidsa
 * Date:   May 30, 2017
 */

int main() {
    int numPeople;      // Number of people that will be eating
    int totalOuncesPasta; // Total ounces of pasta to serve numPeople

    // Get number of people
    cout << "Enter number of people: " << endl;
    cin  >> numPeople;

    // Calculate and print total ounces of pasta
    totalOuncesPasta = numPeople * 3; // Typical ounces per person
    cout << "Cook " << totalOuncesPasta << " ounces of pasta." << endl;

    return 0;
}
```

Feedback?

1.4.1: Comments.

Indicate which are valid comments.

- `// Get user input`
 - ☐ Valid
 - ☐ Invalid
- `/* Get user input */`
 - ☐ Valid
 - ☐ Invalid
- `/* Determine width and height,
 calculate volume,
 and return volume squared.
 */`
 - ☐ Valid
 - ☐ Invalid
- `// Print "Hello" to the screen //`
 - ☐ Valid
 - ☐ Invalid
- `// Print "Hello"
Then print "Goodbye"
And finally return.
//`
 - ☐ Valid
 - ☐ Invalid
- `/*
 * Author: Michelangelo
 * Date: 2014
 * Address: 111 Main St, Pacific Ocean
 */`
 - ☐ Valid
 - ☐ Invalid
- `// numKids = 2; // Typical number`
 - ☐ Valid
 - ☐ Invalid
- `/*
 numKids = 2; // Typical number
 numCars = 5;
 */`
 - ☐ Valid
 - ☐ Invalid
- `/*
 numKids = 2; /* Typical number */
 numCars = 5;
 */`
 - ☐ Valid
 - ☐ Invalid

Feedback?

Whitespace

Whitespace refers to blank spaces (space and tab characters) between items within a statement and blank lines between statements (called newlines). A compiler ignores most whitespace.

Good practice is to deliberately and consistently use whitespace to make a program more readable. Programmers usually follow conventions defined by their company, team, instructor, etc., such as:

- Use blank lines to separate conceptually distinct statements.
- Indent lines the same amount.
- Align items to reduce visual clutter.
- Use a single space before and after any operators like `=`, `+`, `*`, or `<<` to make statements more readable.

Figure 1.4.2: Good use of whitespace.

```
#include <iostream>
using namespace std;

int main() {
    int myFirstVar;    // Aligned comments yield less
    int yetAnotherVar; // visual clutter
    int thirdVar;

    // Above blank line separates variable declarations from the rest
    cout << "Enter a number: ";
    cin  >> myFirstVar;

    // Above blank line separates user input statements from the rest
    yetAnotherVar = myFirstVar; // Aligned = operators
    thirdVar      = yetAnotherVar + 1;
    // Also notice the single-space on left and right of + and =
    // (except when aligning the second = with the first =)

    cout << "Final value is " << thirdVar << endl; // Single-space on each side of <<
    return 0; // The above blank line separates the return from the rest
}
```

Feedback?

Figure 1.4.3: Bad use of whitespace.

```
#include <iostream>
using namespace std;

int main() {
    int numPeople; int    totalOuncesPasta;
    cout<<"Enter number of people: ";cin>>numPeople;
    totalOuncesPasta = numPeople * 3; cout << "Cook " << totalOuncesPasta << " ounces of pasta." << endl;
    return 0;
}
```

Feedback?

1.4.2: Whitespace.

Are the specified lines of code good or bad uses of whitespace?

```
#include <iostream>
using namespace std;

int main() {
    int userAge;
    int currentDecade;
    int  nextDecade;
    int nextMilestone;

    cout << "Enter your age: " << endl;
    cin  >> userAge;

    currentDecade=userAge/10;
    nextDecade = currentDecade + 1;
    nextMilestone = nextDecade * 10;
    cout << "Next big birthday is at " << nextMilestone << endl;

    return 0;
}
```

- `int nextDecade;`
 - ☐ Good
 - ☐ Bad
- `currentDecade=userAge/10;`
 - ☐ Good
 - ☐ Bad
- `nextDecade = currentDecade + 1;`
 - ☐ Good
 - ☐ Bad
- `nextMilestone = nextDecade * 10;`
 - ☐ Good
 - ☐ Bad

Feedback?

Compiling code with comments and whitespace

The animation below provides a (simplified) demonstration of how a compiler processes code from left-to-right and line by line, finding each statement (and generating machine code using 0s and 1s) and ignoring whitespace and comments.

1.4.3: A compiler scans code line by line, left to right; whitespace is mostly irrelevant.

Start ☐ 2x speed

```
cout << 20 + 40 + 50;
// 20 dollars/hr * 40 hrs/wk * 52 wks/year
cout << "Enter integer: ";
cin >> userNum;
```

Compiler ↓

```
1011001101101111
001110101011101
000001101101011
11110001110100
101101100111000
011010101101110
```

High-level program
myfile.cpp

Executable program
myfile.exe

Captions ^

- The compiler converts a high-level program into an executable program using machine code (0s and 1s).
- Comments do not generate machine code.
- The compiler recognizes end of statement by semicolon `;`.

Feedback?

1.4.4: Compiling code with whitespace and comments.

- Spaces are always ignored by the compiler.
 - ☐ True
 - ☐ False

- How many spaces will the compiler ignore in the code below?

```
numToBuy = numNeeded - numInStock + 2;
```

- ☐ 3
- ☐ 6
- ☐ 7

- How many lines will the compiler ignore in the code below?

```
int userAge;
int currentDecade;
int  nextDecade;
int nextMilestone;

// FIXME: Get user age
userAge = 29; // Testing with 29

currentDecade = userAge / 10;
nextDecade = currentDecade + 1;
nextMilestone = nextDecade * 10;
```

- ☐ 1
- ☐ 2
- ☐ 3

Feedback?

1.4.1: Comments and whitespace.

620880 5010016 q3bzp/

Start

1

On line 10, after the semicolon `;`, write a single-line comment that:

- uses `//`
- contains the words `Converting quarts to cups`

Do not use `/*` and `*/`.

```
1 #include "testcode.h" // For testing purposes
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     int quarts;
7     int cups;
8
9     quarts = 6; // 6 quarts is given
10    cups = 4 * quarts;
11
12    Testcode(); // For testing purposes
13
14    return 0;
15 }
```

1

2

Check

Next level

Feedback?

How was this section?



Provide section feedback

Activity summary for assignment: CSC108 CH01 PA

Due: 01/30/2025, 11:59 PM EST

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See [this article](#) for more info.

106 / 106 points

106 / 106 points submitted to Blackboard learn

Completion details