

# CSCI 1380

JANUARY 23, 2016

# THE PROGRAMMING ENVIRONMENT

◀ back to edit

run ▶



Due: Jan 26, 2017 09:59 pm

submit

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!";
7 }
```

# REPL.IT

gcc version 4.6.3

Instructions from your teacher:

## INSTRUCTIONS

Welcome to [repl.it](#) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). With a partner:

Carefully step through the tasks listed below to familiarize yourselves with the programming environment.

As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).

When you are finished, hit "submit" (top right) to affirm that you went through the example. Both you and your partner should do so on your individual [repl.it](#) accounts.

## TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](#), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

## 2. SYNTAX & COMPILATION ERRORS:

**A.** Delete the semicolon from the end of line 6 in the editor. Now try to compile and run the code. What happens?

NOTE: This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp::7::1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it's](#) "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

DO: Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

◀ back to edit

run ▶



Due: Jan 26, 2017 09 : 59 pm

submit

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!";
7 }
```

gcc version 4.6.3

Instructions from your teacher:

### INSTRUCTIONS

Welcome to [repl.it](https://repl.it) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). **With a partner:**

- Carefully step through the tasks listed below to familiarize yourselves with the programming environment.
- As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).
- When you are finished, hit "submit" (top right) to affirm that you went through the example. Both you and your partner should do so via your individual [repl.it](https://repl.it) accounts.

### TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](https://repl.it), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

### 2. SYNTAX & COMPILATION ERRORS:

**A. Delete the semicolon from the end of line 6 in the editor.** Now try to compile and run the code. What happens?

NOTE: This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp::7::1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it](https://repl.it)'s "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

DO: Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

◀ back to edit

run ▶

Due: Jan 26, 2017 09:59 pm

submit

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!";
7 }
```

gcc version 4.6.3

Instructions from your teacher

## INSTRUCTIONS

Welcome to [repl.it](https://repl.it) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). **With a partner:**

- **Carefully step through the tasks listed below to familiarize yourselves with the programming environment.**
- **As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).**
- **When you are finished, hit "submit" (top right) to affirm that you went through the example.** Both you and your partner should do so via your individual [repl.it](https://repl.it) accounts.

## TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](https://repl.it), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

## 2. SYNTAX & COMPILATION ERRORS:

**A. Delete the semicolon from the end of line 6 in the editor.** Now try to compile and run the code. What happens?

NOTE: This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp::7::1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it](https://repl.it)'s "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

DO: Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

◀ back to edit

run ▶



Due: Jan 26, 2017 09:59 pm

submit

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!";
7 }
```

gcc version 4.6.3

Instructions from your teacher:

### INSTRUCTIONS

Welcome to [repl.it](#) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). **With a partner:**

- Carefully step through the tasks listed below to familiarize yourselves with the programming environment.
- As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).
- When you are finished, hit "submit" (top right) to affirm that you went through the example. Both you and your partner should do so via your individual [repl.it](#) accounts.

### TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](#), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

### 2. SYNTAX & COMPILATION ERRORS:

**A.** Delete the semicolon from the end of line 6 in the editor. Now try to compile and run the code. What happens?

**NOTE:** This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp::7::1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it's](#) "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

**DO:** Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

◀ back to edit

run ▶



Due: Jan 26, 2017 09:59 pm

submit

Instructions from your teacher:

### INSTRUCTIONS

Welcome to [repl.it](#) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). **With a partner:**

- Carefully step through the tasks listed below to familiarize yourselves with the programming environment.
- As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).
- When you are finished, hit "submit" (top right) to affirm that you went through the example. Both you and your partner should do so via your individual [repl.it](#) accounts.

### TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](#), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

### 2. SYNTAX & COMPILATION ERRORS:

**A.** Delete the semicolon from the end of line 6 in the editor. Now try to compile and run the code. What happens? NOTE: This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp::7::1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it's](#) "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

**DO:** Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

gcc version 4.6.3



◀ back to edit

run ▶



Due: Jan 26, 2017 09 : 59 pm

submit

Instructions from your teacher:

### INSTRUCTIONS

Welcome to [repl.it](https://repl.it) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). **With a partner:**

- Carefully step through the tasks listed below to familiarize yourselves with the programming environment.
- As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).
- When you are finished, hit "submit" (top right) to affirm that you went through the example. Both you and your partner should do so via your individual [repl.it](https://repl.it) accounts.

### TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](https://repl.it), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

### 2. SYNTAX & COMPILATION ERRORS:

**A.** Delete the semicolon from the end of line 6 in the editor. Now try to compile and run the code. What happens? NOTE: This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp::7::1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it](https://repl.it)'s "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

**DO:** Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

gcc version 4.6.3





◀ back to edit

run ▶



Due: Jan 26, 2017 09 : 59 pm

submit

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!";
7 }
```

gcc version 4.6.3

Instructions from your teacher:

### INSTRUCTIONS

Welcome to [repl.it](#) -- this is the programming environment that we'll be using! This window contains information and instructions provided by the instructor. To the left, we have an editor (top) and console (bottom).

The editor -- similar to other text editors with which you may be familiar (e.g., Notepad, Microsoft Word, etc.) -- is for editing. This editor is specifically designed for editing code.

The console, on the other hand, is like a programming assistant. It is where we will receive error reports from the compiler/computer (if there are errors in our code when we try to run it). It is also where we can send program output and communicate dynamically with the computer.

C++ source code (comprising a simple computer program) has been provided for you to work with (see top left window). **With a partner:**

- Carefully step through the tasks listed below to familiarize yourselves with the programming environment.
- As you are going through the tasks, take a minute to discuss what happens and why (see NOTES).
- When you are finished, hit "submit" (top right) to affirm that you went through the example. Both you and your partner should do so via your individual [repl.it](#) accounts.

### TASKS

**1. COMPILING & RUNNING:** Click the "run" button (top, left; above the editor window) to *compile and run/execute* the program. A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (in our case: C++) into another computer language that is readable by the machine (cf. <https://en.wikipedia.org/wiki/Compiler>). Once the source code has been compiled into an executable program, it can be run. Here in [repl.it](#), clicking the "run" button causes a compiler to attempt to compile the C++ source code, and then, if it compiles, the program is run/executed.

**DISCUSS:** What happened when you compiled/ran the provided code?

### 2. SYNTAX & COMPILATION ERRORS:

**A.** Delete the semicolon from the end of line 6 in the editor. Now try to compile and run the code. What happens?

**NOTE:** This is called a syntax error. A semicolon character is required to terminate C++ statements. If you forget (or remove) a semicolon, the compiler will "throw a syntax error" -- meaning it will print a message to the console indicating the code failed to compile and where ("main.cpp:7:1" indicates there is an error at or before character 1 on line 7 in a file called main.cpp). Note also that the program did not run. When there is a syntax error, [repl.it's](#) "run" function here quits after the compiler returns an error message, without trying to run/execute the program.

**DO:** Replace the semicolon that you deleted. Make sure the code compiles and runs successfully before moving onto

# INTRODUCTION TO C++

**C++** is an *artificial* language.

(For comparison, English, Spanish, French, etc. – are referred to as *natural* languages.)

A **statement** in C++ is like a sentence in natural language.

(Except C++ statements are terminated by **semicolons**.)

# C++ OPERATORS

To tell the computer to do something, we can use something called an **operator**.

There are a number of operators, with each specifying a particular operation...

Addition operator (+): *add two things.*

Subtraction operator (-): *subtract one thing from another.*

Multiplication operator (\*): *multiply two things.*

Etc.

# THE PRINT STATEMENT

A handy operator to know: <<

This is the **insertion operator**. This instructs the computer to print something, somewhere. For example:

```
cout << "Hello!";
```

# THE PRINT STATEMENT (CONT.)

We can print more than just **string literals**. For example:

```
cout << 5;
```

# THE PRINT STATEMENT (CONT.)

Alternatively, instead of `cout << 5;`, we could write:

```
cout << 2 + 3;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 1 + 2 + 3 - 4;
```



# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 1 + 2 + 3 - 4;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 1 + 2 + 3 - 4;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 3 + 3 - 4;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 6 - 4;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 2 ;
```

# COMBINING OPERATIONS (CONT.)

Given the following statement, what will the computer do?

```
cout << 110 - 10 << 0;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 110 - 10 << 0;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 100 << 0;
```



# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 100 << 0;
```

# COMBINING OPERATIONS

Given the following statement, what will the computer do?

```
cout << 100 << 0;
```

VARIABLES

# ADDRESSING

**Addressing** refers to the storage and retrieval of data via a numeric *address*.

A computer with 1GB of RAM has ~one billion locations to store data. Since locations where data can be stored are *sequential*, the first location starts at address 0. The last location is around address 999,999,999.

# ADDRESSING (CONT.)

What if you owned a storage facility with one billion lockers? Imagine trying to keep track of what went where, with just your memory...

# ADDRESSING (CONT.)

We're in a similar situation...When we program computers, we have to keep track of the data that we have/want to retrieve.

This becomes easier if we *name* locations. For example, instead of ~"*store this data at location 986,247,124*", we can tell the computer to "*store this data at location x*".

# VARIABLES

Before a program stores a piece of data, it must allocate space in memory. This involves reserving space at some location in memory and giving that location a *name*.

A **variable** is just that – a named, reserved address in memory.

# VARIABLE DECLARATION

To reserve space in memory for a piece of data, we make a **variable declaration** statement.

Variable declarations are comprised of a **type** and a name (**identifier**).

For example:

```
int x
```



# DATA TYPES

In C++, every variable must have a type (it tells the computer how much space to reserve and how to process the data).

Some C++ data types:

`int`: integer (e.g., 1)

`float`: real number (e.g., 1.0)

`char`: character (e.g., `h`)

`string`: string of characters (e.g., "hello")

# IDENTIFIERS

There are some rules about naming our variables...Identifiers can consist of letters, digits, and the underscore character (\_). However, they **must begin with a letter or underscore.**

For example:

`first`, `First`, `_1st`, and `f_1_r_s_t` are all valid identifiers

# THE READ STATEMENT

Another handy operator to know: >>

This is the **extraction operator**. This instructs the computer to extract some data from somewhere and to save it in memory. For example:

```
string s;  
cin >> s;
```

# A SIMPLE PROGRAM

```
string s;
```

```
cout << "Hello! What is your name?";
```

```
cin >> s;
```

```
cout << "Nice to meet you, " << s;
```

# PARTICIPATION ACTIVITIES

# GROUP COLLABORATION

Must fill out pre-questionnaire to receive group assignment.

Purpose is to learn-by-teaching (1370) and provide peer resources (1380).

First group meeting due by this Thursday (Jan. 26) 8pm.

# ASSIGNMENTS

This week's exercises (due by this Friday, 10pm) posted to repl.it.

Start with group exercises (Labs 1a-d; posted in lab classroom) first. Then move on to assignments (Week 2 Exercises; posted in main classroom).