

Outputs and Inputs

1. Outputs

Use the [cout](#) command to output something to the console screen. The input and output commands are defined in the [iostream](#) library. So in order to include and use this library your program needs to contain the following two lines

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

This is the skeleton template for all your C++ programs.

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

int main () {
    // This is a comment. Write your code here

    return 0;
}
```

The cout command is used in conjunction with the insertion operator, which is written as << (two "less than" signs). For example:

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

int main () {
    // This is a comment. Write your code here
    cout << "Hello World";    // prints the words Hello World on the console

    return 0;
}
```

The insertion operator (<<) may be used more than once in a single cout statement.

```
cout << "Hello " << "World";    // prints the words Hello World on the console
```

cout does not add a new line break after its output unless we explicitly indicate it using the new line keyword [endl](#) or [\n](#) (backslash, n). For example,

```
cout << "First line.";
cout << "Second line.";
cout << "Third line.";
```

will produce the following output

```
First line.Second line.Third line.
```

Whereas, this

```
cout << "First line." << endl;  
cout << " Second line." << endl;  
cout << "Third line." << endl;
```

will produce this output

```
First line.  
Second line.  
Third line.
```

The following will give the same outputs

```
cout << "First line." << endl << "Second line." << endl << "Thrid line." << endl;
```

and

```
cout << "First line.\nSecond line.\nThrid line.\n";
```

and

```
cout << "First line." << endl;  
cout << "Second line.\nThird line." << endl;
```

and

```
cout << "First line." << endl << "Second line.\nThird line." << endl;
```

and

```
cout << "First line." << endl << "Second line." << endl << "Third line." << endl;
```

To print out the double quote use \" like

```
cout << "He said, \"I like apples\".";
```

Questions

1. There are 11 errors in the following program. Find them.

```
include (iostream)
using namespace std

Main() {
    cout << "The works of Mozart/ninclude the following";
    cout << "\nThe Turkish March" endl
    cout << "and "Symphony" No. 40 ";
    //cout << "in G minor." << endl;
    return 0;
}
```

2. Exactly what will the following program print out on the console?

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

int main() {
    cout << "The works of Mozart\ninclude the following";
    cout << "\nThe Turkish March" << endl;
    cout << "and Symphony No. 40 ";
    //cout << "in G minor." << endl;
    return 0;
}
```

3. Write a C++ program that will display your name on the first line, your street address on the second line, your city, state, and ZIP code on the third line, and your email address on the fourth line. Place a comment with your name and today's date at the top of the program. Test your program by compiling and running it. The output will look like this.

Dr. Enoch Hwang
4500 Riverwalk Parkway
Riverside, CA 92505
ehwang@lasierra.edu

4. Write a C++ program to print out a 10×10 square using asterisks.

2. Outputting Math Calculations and Variables

In addition to being able to print out strings (texts), you can also use the [cout](#) command to:

- Print out numbers
- Print out the results from math calculations
- Print out the value stored inside variables. Variables are storage locations for storing values. Every variable must have a unique name. Before you use a variable, you must first declare it.

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

int main () {
    cout << "Hello World";    // prints the words Hello World on the console
    cout << 120;              // prints the number 120 on the console
    cout << 2 + 3;            // addition prints the number 5 on the console
    cout << 6 - 1;           // subtraction
    cout << 8 * 4;           // multiplication
    cout << 2 * 4 - 1;
    cout << 2 * (4 - 1);

    int x = 4;                // declares a variable call x and stores the number 4 into it
    cout << x;                // prints the content of variable x on the console
    cout << x + 7;            // prints the result of adding 7 to whatever value is in x

    return 0;
}
```

Output from the above program:

```
Hello World120553276411
```

Divisions

```
cout << 6 / 2 << endl;      // integer division
cout << 7 / 4 << endl;      // integer division
cout << 7.0 / 4 << endl;    // floating point division
cout << 8 % 3 << endl;      // remainder of division
```

Output from the above program:

```
3
1
1.75
2
```

3. Inputs

Use the [cin](#) command to input something from the user into your program. Notice that the arrows point in the other direction (>>). The arrows are followed by a variable which will be used to store the data that you typed in.

```
int age;  
cin >> age;
```

The cin command processes the input from the keyboard only after the Enter key has been pressed. Note that on the Mac keyboard, you have to use the Return key on the main keyboard and NOT the Enter key on the numeric keypad.

You can also use cin to request more than one input from the user:

```
int a, b;  
cin >> a >> b;
```

is equivalent to:

```
int a, b;  
cin >> a;  
cin >> b;
```

In both cases the user must enter two values, one for variable *a* and another one for variable *b* that are separated by a space, a tab character or the Enter key. For each variable specified in the cin command, it will read characters from the keyboard until a white space or the Enter key is reached. So if you execute the following code and enter John Doe as the input

```
string name;  
cout << "Enter your name? ";  
cin >> name;  
cout << "Your name is " << name << endl;
```

you will be surprised that only the first name John is read as seen next:

```
Enter your name? John Doe  
Your name is John
```

In order to get entire lines (including white spaces), we can use the function [getline](#), which gets the entire line with all the white spaces up to the Enter key.

```

string name;
string age;

cout << "Enter your name? ";
getline (cin, name);
cout << "Your name is " << name << endl;
cout << "How old are you? ";
getline (cin, age);
cout << "You are " << age << " years old!" << endl;

```

Notice in the above example when you enter your age, the number is stored as a string and not as a number. If you need to do some calculations with that number, you will first have to convert the string to an actual number. There are different ways of doing the conversion: [stoi](#), [stof](#), and [stringstream](#) as shown next.

```

#include <iostream>
#include <sstream> // need to include this library to use stringstream
using namespace std;

int main ( ) {
    string mystring;
    float price;
    int quantity;

    cout << "Enter price: ";
    getline (cin, mystring);
    price = stof(mystring); // convert mystring to a float
    stringstream (mystring) >> price; // another way to convert mystring to a number

    cout << "Enter quantity: ";
    getline (cin, mystring);
    quantity = stoi(mystring); // convert mystring to a number
    stringstream(mystring) >> quantity; // another way to convert mystring to a number

    cout << "Total price: " << price * quantity << endl; // notice the * is used for multiplication
    return 0;
}

```

Alternatively, you can directly enter the numbers into an integer or float variable instead of to a string variable, so that no conversion is necessary as shown next:

```

#include <iostream>
using namespace std;

int main ( ) {
    float price;
    int quantity;

    cout << "Enter price: ";
    cin >> price;
    cout << "Enter quantity: ";
    cin >> quantity;
    cout << "Total price: " << price * quantity << endl;
}

```

```
return 0;  
}
```

4. Formatting Outputs

You can also tell cout to display numbers in different formats such as how many digits after the decimal point. For example

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

int main() {
    float price;
    price = 78.5;
    cout << "The price is $" << price << endl;

    return 0;
}
```

will print out:

```
The price is $78.5
```

Whereas, this

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

int main() {
    float price;
    price = 78.5;
    cout.setf(ios::fixed);           // use fixed format
    cout.setf(ios::showpoint);      // use a decimal
    cout.precision(2);              // two digits after the decimal point
    cout << "The price is $" << price << endl;

    return 0;
}
```

will print out:

```
The price is $78.50
```

Notice that the ending zero is automatically added.

You can also use `setw` within the `cout` command to specify how many spaces to use in the printout. The string after the `setw` will be right justified in the allotted spaces. To use `setw`, you need to include the `iomanip` library.

```
#include <iostream>
#include <iomanip>      // need to include iomanip to use setw
using namespace std;

int main() {
    cout << "12345678901234567890" << endl;
    // reserve 12 spaces for the name and 8 spaces for the age
    cout << setw(12) << "Name" << setw(8) << "Age" << endl;
    cout << setw(12) << "Jonathan" << setw(8) << "18";

    return 0;
}
```

The above program will print out:

```
12345678901234567890
      Name      Age
Jonathan      18
```

5. Optional stuff: printf and scanf (extra for backward compatibility)

You can also use the [printf](#) command to output to the screen and [scanf](#) command to input from the keyboard¹. These two commands are for backward compatibility, i.e. for old C programs.

The following program will first ask you to type in your name and your age, and after you have typed in your name and age, it will print out “Hello” followed by your name and then print out how old you are.

```
char name[20];
int age;

printf("Enter your name? ");
scanf("%s", name);
printf("Enter your age? ");
scanf("%i", &age);
printf("Hello %s, you are %i years old.\n", name, age);
```

A sample output from the program is shown next:

```
Enter your name? Jane
Enter your age? 18
Hello Jane, you are 18 years old.
```

A white space terminates the scanf command. So where it asks for your name and you enter your first and last name separated with a space, you will get the following wrong result instead:

```
Enter your name? Jane Doe
Enter your age? Hello Jane, 0 years old.
```

Notice that the last name is missing in the output and the age is 0. The reason is that the last name Doe is actually assign into the integer variable age, so age gets the value 0.

¹ Technically, these are C commands and not C++ commands.

Notes on using the printf command:

- When printing out a variable, you need to use a formatting character (see table below) as a place holder for that variable. For example, the next command prints out the value in the variable `age` at the location where the `%i` is at within the string.

```
printf("Your age is %i.\n", age);
```

Notes on using the scanf command:

- Need to include a formatting character within double quotes similar to the printf command.
- When entering a string such as your name, the variable cannot be of type string. It must be of type char with the maximum number of characters specified in square brackets as in

```
char name[20];
```

- When entering a number such as your age, you need to have an ampersand sign `&` in front of the variable as in

```
scanf("%i", &age);
```

Notice that there is no `&` used for the printf command as in

```
printf("%i", age);
```

Formatting Characters

| Formatting Character | Output | Example |
|----------------------|----------------------------------------------------------|---------|
| c | Character | a |
| d or i | Signed decimal integer | 392 |
| f | Decimal floating point | 392.65 |
| g | Use the shorter of %f | 392.65 |
| s | String of characters | sample |
| % | A % followed by another % character will print out an %% | % |

Examples with printf

```
printf ("Characters: %c \n", 'a');
printf ("Integers: %i \n", 2013);
printf ("Preceding with blanks: %10i \n", 2013); // use a total of 10 spaces
printf ("floats: %4.2f \n", 3.1416); // use a total of 4 spaces and two digits after
printf ("Width tricks: %*i \n", 5, 10); // use 5 spaces
printf ("%s \n", "A string");
```

And here is the output:

```
Characters: a
Decimals: 2013
Preceding with blanks:      2013
floats: 3.14
Width tricks:    10
A string
```

6. Exercises (Questions with an * are more difficult)

1. This cout line

```
cout << "Hello, " << "I am " << "a C++ statement";
```

is used to print out the message

```
Hello, I am a C++ statement
```

Simplify this cout line (make it shorter) to print out the same message.

2. What happens when you use an integer variable in a cin command and the user enters a decimal number such as 3.1415?
3. What happens when you use an integer variable in a cin command and the user enters a string such as Riverside?
4. Write a program to enter two different kinds of fruits, say “apples” and “oranges,” and then print out the statement “I like to eat apples but not oranges!” Your program needs to replace the words apples and oranges with whatever two fruits the user enters in.
5. Write a program to enter two integer numbers and then print out the sum of these two numbers using the integer format.
6. Write a program to enter four integer numbers and then print out the product of these four numbers using the integer format.
7. Write a program to enter two integer numbers and then print out the quotient of these two numbers (first number divided by the second number). For example, $1/3=0.333$.
8. Write a program to enter the address information for a person, i.e., name, address, city, state and zip code, and then print out the address correctly formatted like this

```
John Doe
123 Magnolia Street
Riverside, CA 92515
```

9. Write a program to enter the names of three people and their telephone numbers, and then print them out one per line nicely formatted in two columns like this

```
Dr Hwang          951-123-4567
John Lopez        456-1390
Ana Smith         789-642-9037
```

Hint: use setw to right-justified the two columns.

10. * Write a program to enter one number and then print out the multiplication table from 1 to 10 for that one number. For example, if you enter 3, then print out $1 \times 3 = 3$, $2 \times 3 = 6$, etc. on separate lines like this

```
1 x 3 = 3
2 x 3 = 6
3 x 3 = 9
...
10 x 3 = 30
```