

CS161: Introduction to Computer Science I

Week 2 (a)

10/2022

Programs are...

- an expression of a series of instructions that the computer must perform
- written in precise languages, called **programming languages**

Programming languages...

- translate your ideas into specific language that the computer will understand
- C, C++, Java, Pascal, Visual Basic, Cobol, Fortran

Solving Problems w/ Computers

The most difficult part of programming is figuring out how to design a method to solve a problem

Only then do we translate this into C++!

Therefore, start by writing an **algorithm** once you understand the problem

An **algorithm** is a sequence of step by step instructions for performing some task -- usually for computation

An **algorithm** must...

- accomplish the task
- be clear and easy to understand
- define the sequence of steps needed to accomplish the task in the order specified

It is best to **ignore** the details of your programming language (C++) when trying to solve a problem

Instead, figure out the steps you need to go thru to solve the problem

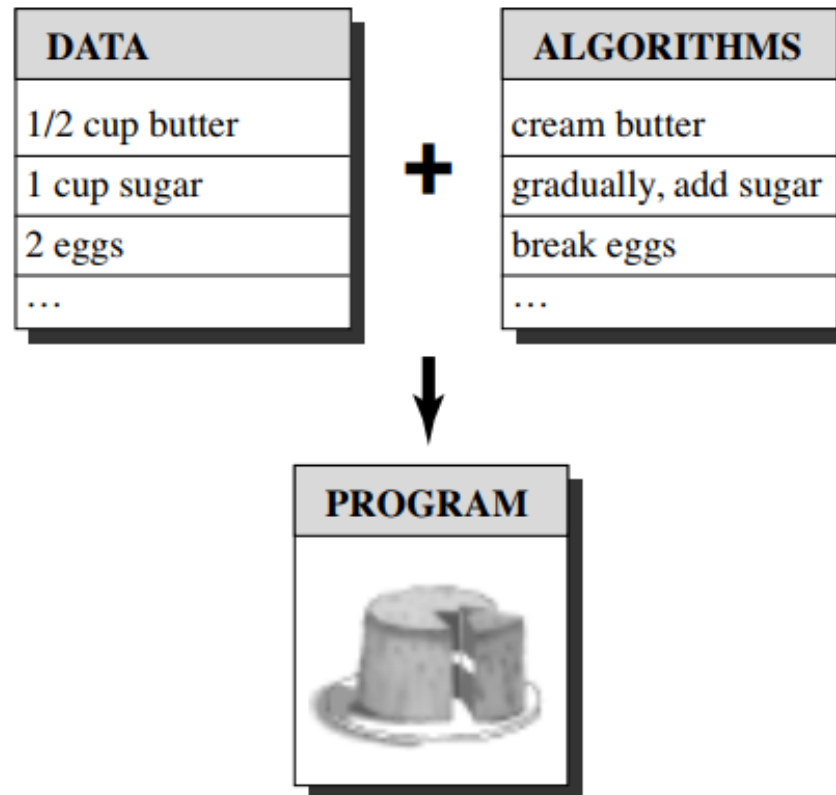
Write these steps down in English

These steps are called the algorithm!

Think of your algorithm as a tool for creating the instructions for solving a problem....as if you were to tell them to another person.

Remember an algorithm is a sequence of step by step instructions for performing some tasks

Solving Problems w/ Computers



For Example...solve a problem

Convert inches to millimeters

- First understand the problem
 - where do the inches come from (the user)
 - what is the math needed for the conversion
 - $\text{mm} = 25.4 \text{ times inches}$
 - how do we want to display the results
 - `2in convert to 50.8mm`

Convert inches to millimeters

Next, write the algorithm

- **Step 1:** Welcome the user
tell them what to expect
tell them the purpose of the program

Convert inches to millimeters

Step 2:

- Get the number of inches from the user
display a **prompt** asking the user to enter
read in the number of inches
display what was read (**echo**)
ask the user if this is really correct (**confirm**)
if not, repeat this step until the user is satisfied

Convert inches to millimeters

Continuing with **Steps 3** and **4**:

- Convert the number of inches to mm
mm = 25.4 times inches
- Display the results
- Provide a sign-off message

Convert inches to millimeters

The next step is to turn this into a C++ program!
All programs have the following “form”

```
#include <iostream>
using namespace std;
//header comments...
int main()
{
    //program body goes here...
    return 0;
}
```

Convert inches to millimeters

```
#include <iostream>
using namespace std;
// *****
// CS161 Programming Assignment #0
// Purpose of this program is to convert
// inches entered in by the user into
// millimeters and display the results
// *****
int main()
{

}
```

(Different Kind of Comment...)

```
#include <iostream>
using namespace std;
/* *****
   CS161 Programming Assignment #0
   Purpose of this program is to convert
   inches entered in by the user into
   millimeters and display the results
   ***** */
int main() {

}
```

Convert inches to millimeters

```
//Define variables
float inches;    //to save # inches
float mm;        //to save the result

//Step #1, welcome the user
cout <<"Welcome! We will be converting"
      <<" inches to mm today " <<endl;
```

(A different way to do this...)

```
//Define variables
```

```
float inches,          //to save # inches  
      mm;             //to save the result
```

```
//Step #1, welcome the user
```

```
cout <<"Welcome! We will be converting ";  
cout <<" inches to mm today " <<endl;
```

(NOTE: endl is end followed by a letter l)

Convert inches to millimeters

```
//Step #2, Get the input (prompt, read)
cout <<"Please enter the number of inches
"
    <<" that you wish to convert: " ;

cin >> inches; //read the # inches

//echo what was entered
cout <<"You entered: " <<inches <<"in "
    <<endl;
```

Convert inches to millimeters

```
//Step #3 Convert inches to millimeters
```

```
mm = 25.4 * inches;
```

```
//Step #4 Display the results
```

```
cout <<inches <<"in converts to "  
      <<mm <<"mm " <<endl;
```

```
//Step #5 Sign off message
```

```
cout <<"Thank you for using CONVERT "  
      <<endl;
```

```
return 0;
```

```
}
```

Exercises!!!

Write the algorithms (steps in natural languages) to solve the following problems:

1. Given a and b , tell which one is bigger or they are equal.
2. Given a and b , find x which satisfies $ax + b = 0$
3. Given $a \neq 0$, b , and c , find x which satisfies $ax^2 + bx + c = 0$

A graphical representation of steps using various boxes and lines.

Used in analyzing, designing, documenting or managing a process

Flowchart – Common Symbols

Flowline



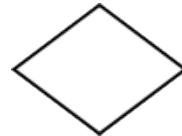
Terminal



Process



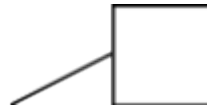
Decision



Input/output



Comment



Predefined process



Draw the flowcharts of the algorithms to solve the problems in page 20 of this slide.

Solving Problems with Computers

- Write a short **Program**
- Basic Form of all C++ programs
- What are Variables, Data Types, Statements
- How do we perform input and output

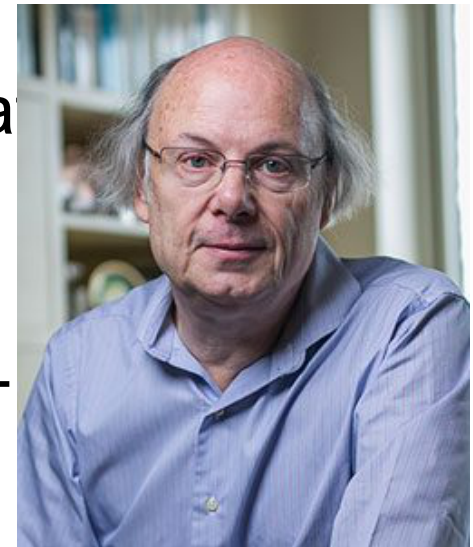
1972: Dennis Ritchie at Bell Labs designs C and 90% of UNIX is then written in C.

Late 70's: OOP becomes popular.

Bjarne Stroustrup at Bell Labs adds features to form "C with classes"

1983: name C++ first used

1998: ISO/ANSI standardization of C++



Bjarne Stroustrup

Convert inches to millimeters

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

// *****
// CS161 Programming Assignment #0
// Purpose of this program is to convert
// inches entered in by the user into
// millimeters and display the results
// *****
int main() {
```

Preprocessor Directives

```
#include <iostream>
```

This is a preprocessor directive

#include allows our programs to copy the contents of some other file (iostream, for example) and have it behave as if that file was typed at the beginning of our programs! iostream allows our programs to perform input from the keyboard (standard in) and output to the screen (standard out)

Header Comments...

```
// *****  
// CS161 Programming Assignment #0
```

These are **in line** comments and provide for documentation in our programs

Once encountered, the rest of the line is taken as documentation.

If you want to surround them by asterisks -- **MAKE SURE** to place at least one space between the // and the asterisks.... otherwise, your entire program will be mistaken as a comment!?!?!

(Different Kind of Comment...)

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

/* *****
   CS161 Programming Assignment #0
   Purpose of this program is to convert
   inches entered in by the user into
   millimeters and display the results
   ***** */

int main() {
}
```

(Different Kind of Comment...)

```
/******  
    Minh H. Nguyen  
******/
```

This type of comment is best used when writing a large block of comments. They begin with a `/*` and end when you type a `*/`

If you forget the ending `*/` your entire program is taken as a comment!!!

I recommend placing a space between the `/*` and any other asterisks....

Variable Definitions

```
//Define variables  
float inches; //to save # inches  
float mm;      //to save the result
```

What are variables?

How are they defined?

What is a data type and why is it important?

What kind of variable names can we use?

Variable Definitions

What are variables?

- Allocate Memory to store data

How are they defined?

data_type variable_name

What is a data type and why is it important?

- **float, int, char, bool** are the fundamental ones
- **double, short, long** are additional ones

What kind of variable names can we use?

- must start with a letter, be any combination of letters, digits, or underscores.

Integer types

- Integers are numbers with no fractional part, such as 15, 0, -157.
- How many numbers are there?
Infinite
- A language can only represent a subset of all integers.

In C++:

- short: 16 bits
- int: 16/32 bits
- long: 32 bits
- long long: 64 bits
- char: 8 bits
- bool: 8 bits

Floating-point Types

- Numbers with fractional part, such as 0.56, 7.85, 128.3333
 - A computer stores such values in 2 parts:
 - Value part
 - Scaling part (up/down)
- Example: 34.12345 and 341234.5:
- Same value part: 0.3412345 (base value)
 - Scaling part: 100 (34.12345) and 100,000 (341234.5)

In C++:

- float: 32 bits
- double: 64 bits
- long double: 128 bits

const Qualifier

- **const type name = value;**

- Example:

```
// Months is symbolic constant for 12
```

```
const int Months = 12;
```

→ The value of Months cannot be changed subsequently.

Convert inches to millimeters

```
int main()
{
    //Define variables
    float inches;    //to save # inches
    float mm;        //to save the result

    //Step #1, welcome the user
    cout <<"Welcome! We will be converting"
         <<" inches to mm today" <<endl;
}
```

Displaying Output

```
cout << "message" <<endl;
```

Pronounced see out

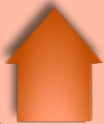
<< is the **insertion operator**

Think of << as an arrow. The message you want to display is being sent to the OUTPUT device in the direction of the arrow:

output_device  message in double quotes

Output after first cout

Welcome! We will be converting inches to mm today



(A different way to do this...)

```
//Define variables
float inches,          //to save # inches
      mm;              //to save the result

//Step #1, welcome the user
cout <<"Welcome! We will be converting";
cout <<" inches to mm today" <<endl;
```

(NOTE: endl is end followed by a letter l)

Convert inches to millimeters



```
//Step #2, Get the input (prompt, read)
cout <<"Please enter the number of
      inches"
      <<" that you wish to convert: ";

cin >> inches; //read the # inches

//echo what was entered
cout <<"You entered: " <<inches <<"in"
      <<endl;
```


Output after 2nd cout

Welcome! We will be converting inches to mm today

Please enter the number of inches that you wish to convert:



Output after 2nd cout

Welcome! We will be converting inches to mm today

Please enter the number of inches that you wish to convert: 2



Output after 2nd cout

Welcome! We will be converting inches to mm today

Please enter the number of inches that you wish to convert: 2

You entered: 2in




Receiving Input

```
cin >> inches;
```

Pronounced see in

>> is the **extraction operator**

Think of >> as an arrow. The data you want to receive is coming from the input device in the direction of the arrow and being saved in memory:

input_device  variable_name

Convert inches to millimeters

```
//Step #3 Convert inches to millimeters
mm = 25.4 * inches;

//Step #4 Display the results
cout <<inches <<"in converts to "
      <<mm <<"mm" <<endl;

//Step #5 Sign off message
cout <<"Thank you for using CONVERT"
      <<endl;
return 0;
}
```

Assignment Operation

//Step #3 Convert inches to millimeters

```
mm = 25.4 * inches;
```

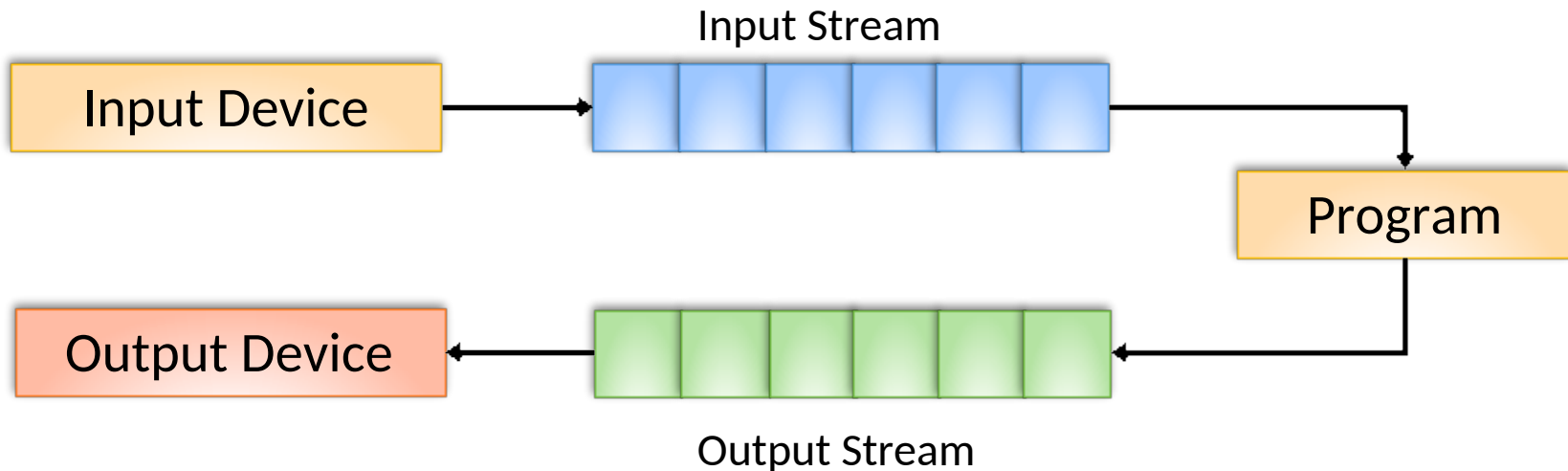
Multiplication requires the asterisk

- can't leave it out like we do in math
- 3x would be written 3 * x in C++
- = is the assignment operation
- takes the value on the right and saves it in the memory for the variable on the left

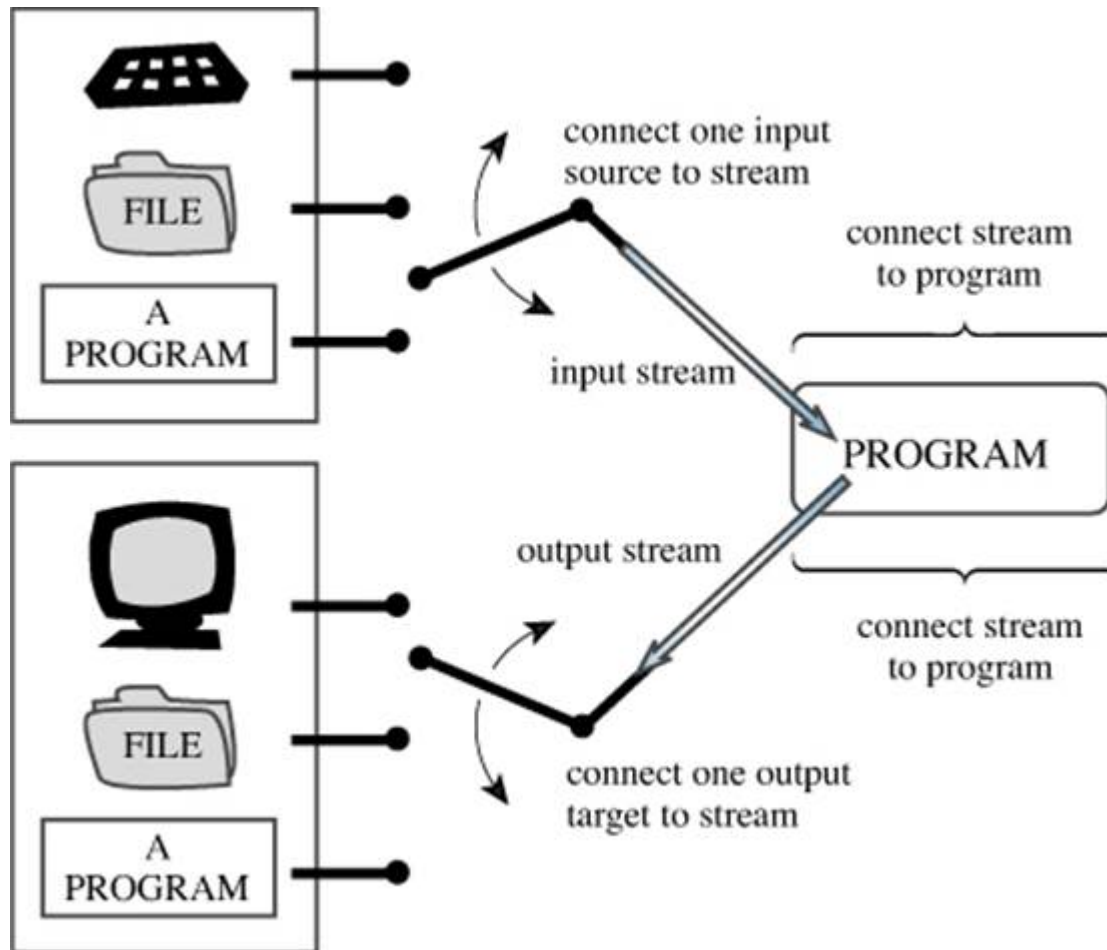
Put the `#include <iostream>` in row #1
`endl` is “e n d” followed by lower case L
spaces inside double quotes ARE important
`cout` vs `cout`
watch your semicolons!

Input Stream: The source stream that provides data to the program

Output Stream: The destination stream that receives output from the program



C++ Stream



We can read integers, floating point numbers and characters using the extraction operator

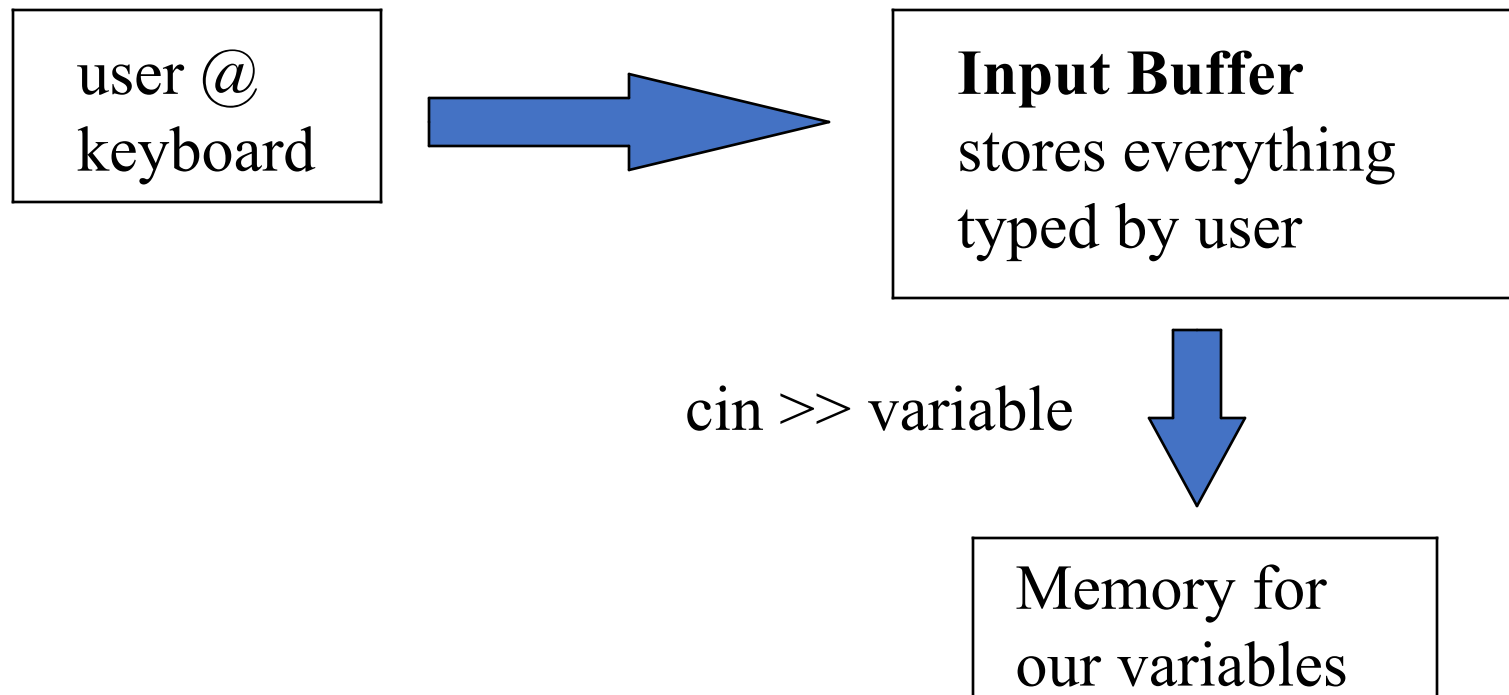
We can't, however, control what the user types in.

Anything the user types in...goes into the input buffer once they hit the enter (or return) key...regardless of what our programs might want!

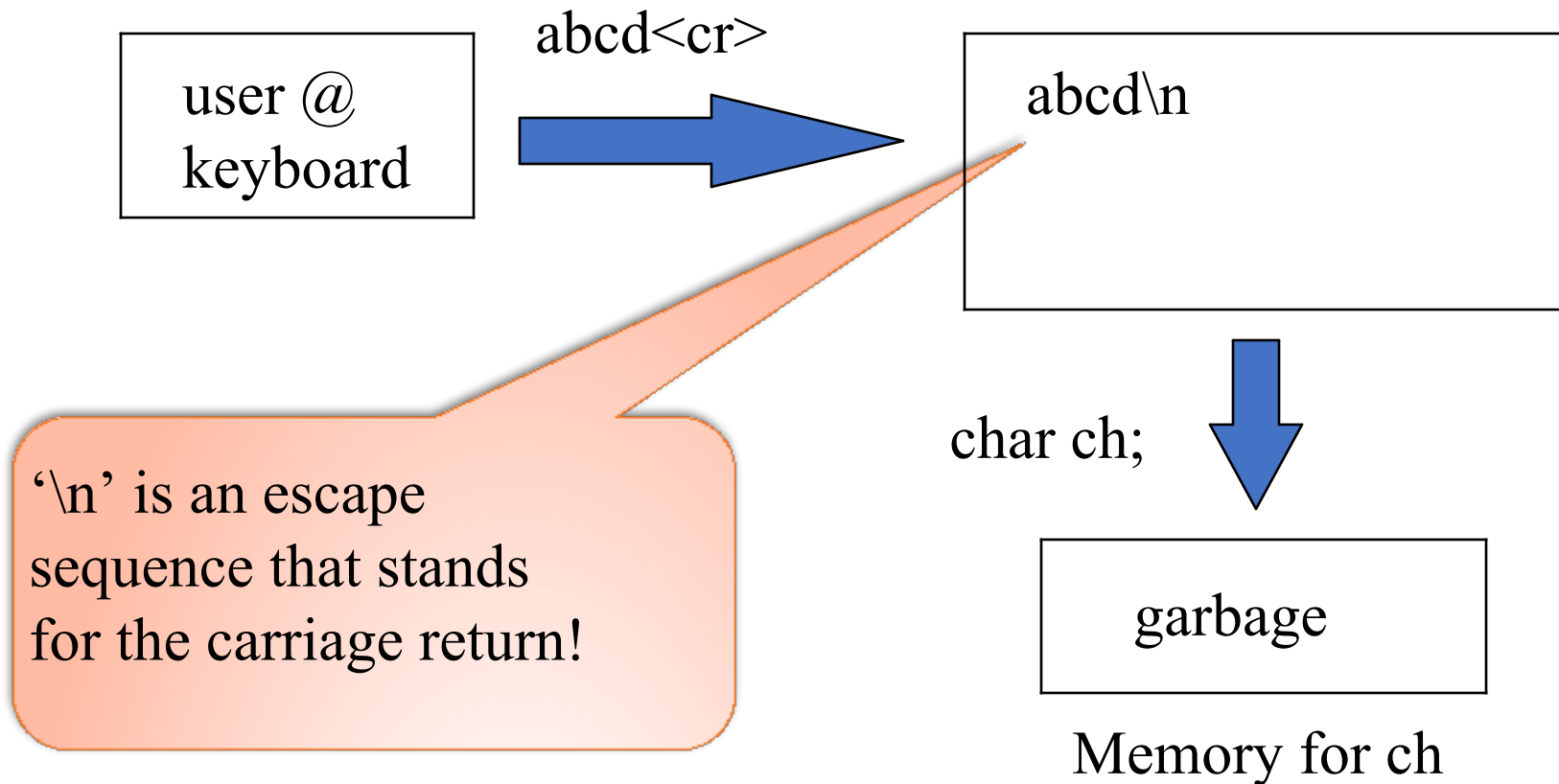
Therefore, it is important to prompt users, so they know exactly what they are supposed to type in

And, it is important to understand how input operations behave

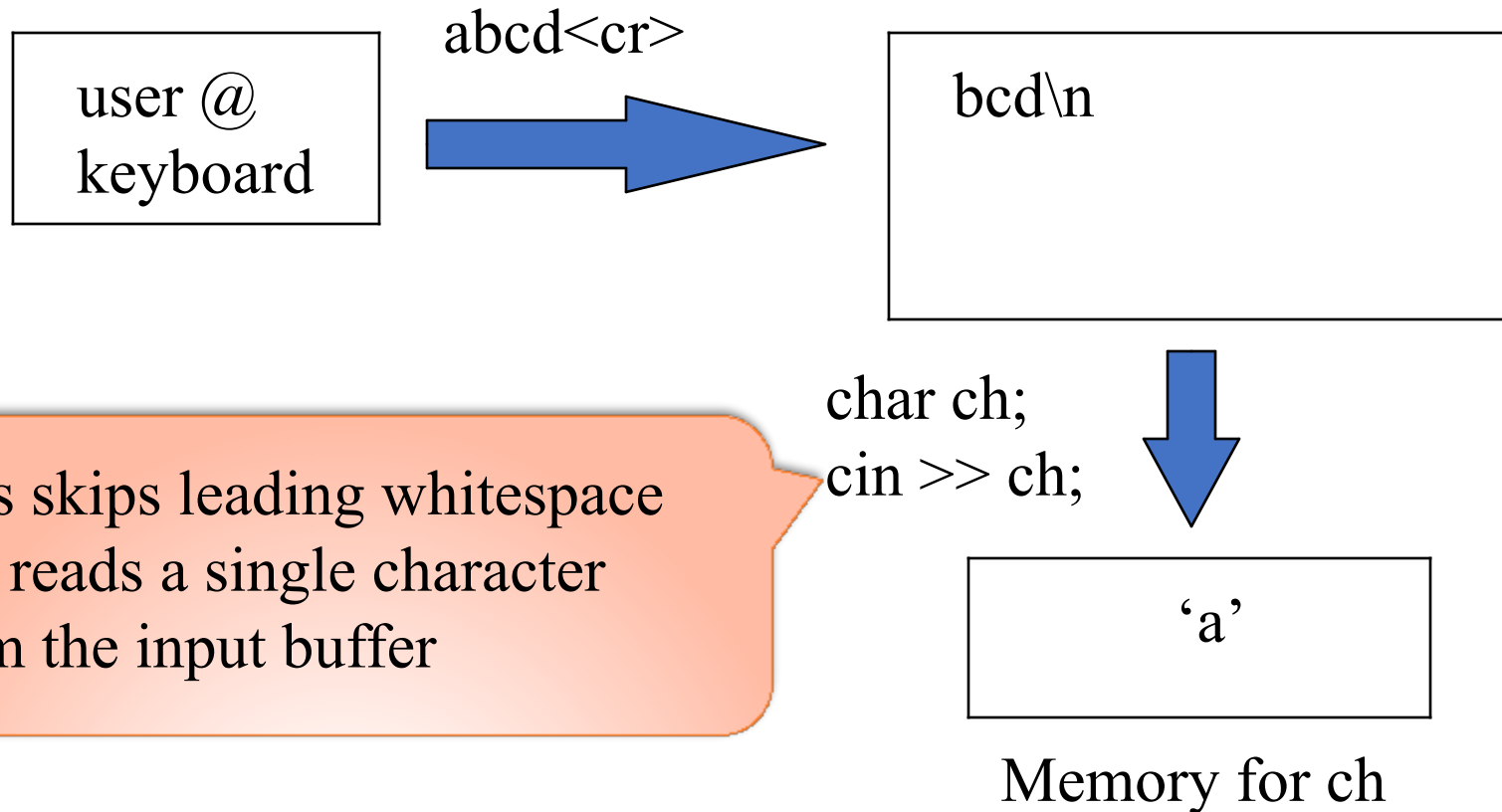
Input Stream



Input Stream



Input Stream



What about integers?

```
int    number;  
cin >> number;
```

Skips leading whitespace and reads in digits until it gets to a non-digit, from the input buffer.

What about floating point numbers?

```
float inches;
```

```
cin >> inches;
```

Skips leading whitespace and reads in digits and optionally one decimal point until it gets to a non-digit or more than one decimal point from the input buffer.

Convert Celcius degree to Farenheit and vice versa.

- Write the algorithm
- Use the program template to write a program

Hints:

- $0^{\circ}\text{C} = 32^{\circ}\text{F}$
- $T(^{\circ}\text{F}) = T(^{\circ}\text{C}) \times 1.8 + 32$
- $T(^{\circ}\text{C}) = (T(^{\circ}\text{F}) - 32) / 1.8$

Any questions???

