* <http://ace.cs.ohiou.edu/new_users/error.html>
* First midterm: Thursday of the fourth week
* First midterm: Thursday of the eighth week
* Final: Saturday, at the end of the tenth week
* Visual C++
* Xcode(clang++/g++)
* What goes into the computer:
* Memory,
* Central processing unit, CPU,
* Input/output device
* Basically moving memory around,
* Controls various devices,
* Input devices:keyboard, temperature sensor sending data to computer
* Output devices: screen
* Memory: store numbers, store value
* H E L L O

ASCII:

H🡺 72

E🡺 101

Space 🡺 32

6 🡺 54

then Unicode

the way to store an audio: take sample and then store each sample as a two dimensional matrix

the way to store a picture: make each pixel represent a number.



24 frames every second, human eyes can’t recognize.

What else can we represent just as memory:

The program itself. Didn’t happen until really recent.

Von Neumann architecture

Cpu loop:

Take instruction count,

Says give me the number at location XXXX

Then circuitry split the number to two pieces,

Operation code and address,

Base on the operation code, the circuitry move things around.

For example, 21 code means copy the number at the indicated memory address into the accumulator.

Takes instruction counter, gives to the logic unit and then do stuff to them.

Very simple operation, takes instruction counter.

21 copy the number at indicated memory address into the accumulator

11: add the number at the indicated memory location to the accumulator

22: copy the number in the accumulator into the indicated memory location

99: halt

first two digit: what kind of operation to perform

other three digits: the address

* Can’t notice a problem really quickly,(problem with machine language programming)
* Assembly language—
* The program is translated into machine language by program called assembler.

An early 1950s development

Every line in assembly language represent either one instruction or one location.

If the line is space, the next will be taken as instruction and the words after this will be taken as address.

e.g.

Load Price

Add Fees

Store Total

Halt

If no space:

Price Data 42

Fees Data 13

Total Data

By restricting the language, it will be easier to write an language.

Because it’s not good enough, people work on a higher level language.

The first one:

Fortran—formula translator:

Integer price=42

Integer fees=13

Integer total=price+less

* + -------so called higher level language
* it doesn’t translate to machine language anymore, but generate machine language code.

Advantage: the program is short, the language might be used for more familiar notation, Easier to read.

--of course, there’s a program that translate Fortran to machine language.

* + Computer cannot understand Fortran, it can only understand machine language.

The program is compiled into machine language by a compiler.

This lead to an explosion of programming language.

Fortran—good for math stuff.

C is a group of language that survived.

Objective-C

C++

Both developed in 1980s, both can take C program and compile. But add enhancement and stuff.

Bigger analogy:

Whole new syntax,

1998 150 C++ standard (c++98)

2015 C++14 due

major compilers—been implementing C++11 for a while. Pretty much has C++ eleven.

Recipe (English) 🡺 Cooking robot 🡺 yummy cake

Correct: Beat 3 eggs into a mixing bowl

Into eggs mixing best 3 bowl a ---compilation error/syntax error (something wrong with the command, misspelled etc.)

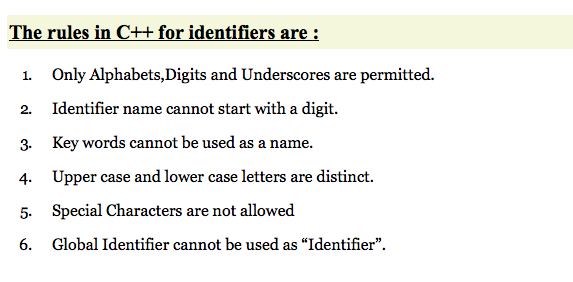
Beat 4000 eggs into a mixing bowl ----logic error/runtime error.

DistanceFromOrigin(x,y)=sqrt(f(x,y,))

DistanceFromOrigin(3.9,7.6)

* Submit the stuff in a single tar file.
* Include<iostream>
* Using namespace std;
* int main()
* {
* }
* namespace group your function in different namespaces.
* Have different nam1espaces in the file. By default, can’t have same name for different variables.
* Int main()
* Char main()
* Void main()
* Containers: x is a container.

|  |  |
| --- | --- |
| char | U |
| Int | y |
| Type | Identifier |

* an integer gets two bits
* double gets four bits
* if it’s a character, it’s one bit.
* Save the first number for sign.
* The compiler for the most part doesn’t care about the spaces between, in another words.
* But spaces in characters matters.
  + A name has to start with letters, use letters or digits or underscores
* C++ is case sensitive
* Do a table that look like this :
* True value what is written
* 1234.56 1234.56
* 1234.5 1234.50
* 1200 1200.00
* 1234.5678 1234.57
* double: about 15 significant digits
* arithmetic expression:
* \* multiplication
* / division, have higher precedence
* c++ requires a type specifier for all declaration
* double---a type name
* double circum() {return 2\*radius\*3.14159265;}
* in this case, circum has an indentifier before that and therefore when called, it needed to be added

|  |  |
| --- | --- |
|  | * Rectangle::Rectangle (int x, int y) : width(x) { height=y; } |

* Member initialization
* ::-----scope operator
* inline member function
* remember that if nothing else is specified, all members of a class defined with keyword class have private access
* Constructors cannot be called explicitly as if they were regular member functions. They are only executed once, when a new object of that class is created.
* constructors with a single parameter can be called using the variable initialization syntax (an equal sign followed by the argument):
* class\_name object\_name = initialization\_value;
* != no equal
* Cannot combine with previous ‘int’ declaration specificer
* Declaration without initialization : type variable;
* Declaration with initialization: type variable =expression;

&& 🡪 and

|| 🡪 or

type-data type:

some common data types are :

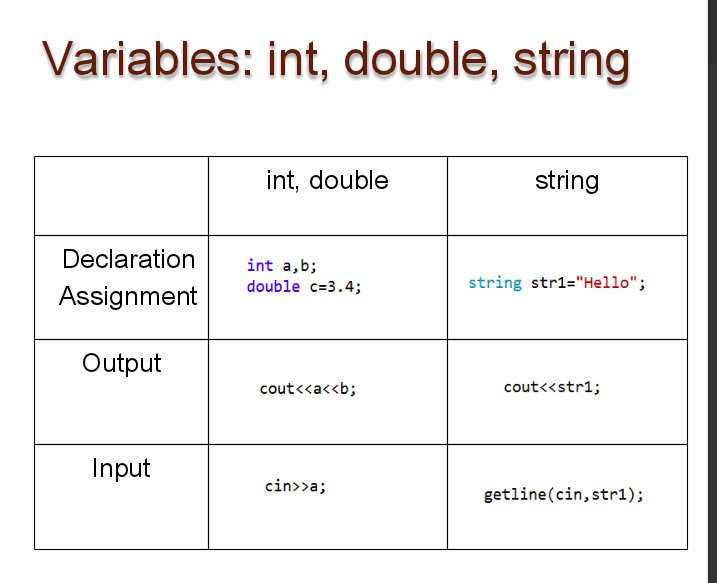
intergers

Booleans

Characters

Floating-point numbers

Alphaunmeric strings



else

the program only recognizes one statement after else. If no brackets

* // istream::ignore example
* #include <iostream> // std::cin, std::cout
* int main () {
* char first, last;
* std::cout << "Please, enter your first name followed by your surname: ";
* first = std::cin.get(); // get one character
* std::cin.ignore(256,' '); // ignore until space
* last = std::cin.get(); // get one character
* std::cout << "Your initials are " << first << last << '\n';
* return 0;

}

variables, naming rules:

don’t start with numbers

no special characters

\_ is ok

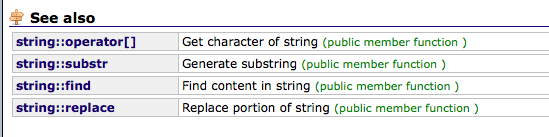
no reserved keywords

start with lower case

* If the number input is not an int you will get a cin.fail(). An error.
* cin.clear() will clear a cin.fail()
* cin.peek() will look at the next character in the buffer. If it is eof() or new line (\n) then that is ok. If not ok then there is extra data in the buffer.
* To clear the buffer then use
* cin.ignore(numeric\_limits <streamsize>::max(), '\n');
* You are clearing the whole of the buffer using limits values.
* The command cin.ignore(1000,'.') will ignore the first 1000 characters or up to the first '.', which ever comes first.
* // istream::getline example
* #include <iostream> // std::cin, std::cout
* int main () {
* char name[6], title[6];
* std::cout << "Please, enter your name: ";
* std::cin.getline (name,6);
* std::cout << "Please, enter your favourite movie: ";
* std::cin.getline (title,6);
* std::cout << name << "'s favourite movie is " << title;
* return 0;
* }
* get the first six characters
* cout.setf(ios::fixed) // "fixed" is a format option that displays floating point numbers in normal notation - no trailing zeroes and no scientific notation.
* x+=5;
* x-=5;
* x/=5;
* x\*=5;
* a=5, a\*=4+6, a=50🡪 pay attention to the sequence of the algorithm
* if (a=b), always yes.
* “\t” 🡪 is space
* cout.sef(ios::showpoint)🡪 shows the decimal point;
* cout.precision(n)-🡪 the precision is n (an integer)
* if use const for a variable and then give a value to this variable, the compiler would say:
* “No viable overload ‘=’ ”
* else if🡪 if not the case, skip rest of the block on go into the bottom part
* getline(cin,string) is only for the data type string, not char. Char is only one.
* testing an integer
* a switch statement:
* switch (choice)
* {
* case 1:
* …
* break;
* case 2:
* ….
* break;
* }
* ---switch must be integer type
* --the most common error: left off the break.
* --the final break is optional.
* Switch expression can’t be double .
* Tyr to make progress to make the expression wrong. So that the whole thing don’t run forever.
* For example, the controlling software for a microwave is an infinite loop.
* Do
* {…….
* }

while (…);

for (initialization: stay-in-loop condition; prepare-for-next-iteration)

* exit the loop”
* for ..
* for..
* for…
* statement1;
* statement 2;
* statement 2 will be excluded from the for loop if without bracket.
* s[k] and s.at(k) is the same to find out the position of a character in a string.
* For loop:
* If there’re no brakets, it automatically recognize the next line as the only statement in the loop.
* #include <iostream>
* using namespace std;
* int main()
* {
* for( ; ; ) ;
* return 0;
* }
* this is an infinite loop
* size\_t size() const;Return length of string Returns a count of the number of characters in the string. string::length is an alias of string::size, returning both the exact same value.
* Accessing a character at a position not in the range 0 through s.size()-1 is a string s is undefined behavior. //output a inverse of a ?
* 
* “\t” the tab character
* “\’ ” the ‘ character
* “\\” the \ character
* #include <iostream>
* using namespace std;
* int main ()
* {
* char ch;
* cin >> ch;
* if (isdigit(ch))
* {
* cout<< "the character is a dignit \n";
* }
* if (islower(ch))
* {
* cout << " the charac is a lower case alphabet \n";
* }
* if (isupper(ch))
* cout << "the characer is an upper character \n"
* }
* isXXX is a Boolean expression that look up whether the character is certain type.
* Toupper(x) makes x an upper case letter.
* Tolower(X) makes X an lower case letter.
* Build your own function:
* void greet()
* {
* for (int k=1;k<3;k++)
* {
* cout << "Hello" << endl;
* }
* }
* #include <iostream>
* using namespace std;
* int main() {
* int i , j;
* for( i = 0 ; i < 2 ; i++ )  
  for( j = 0 ; j < 2 ; j++ )
* cout << i \* j; cout << endl;
* return 0; }

when it moves to the upper loop, the lower loop refreshes

isalpha—is alphabetical?

* books:
* indentifiers—name of a variable
* C++🡪 case sensitive
* Literals—the name of a specific value
* Const int---const🡪 modifier
* double ans = n/static\_cast<double>(m);
* cout🡪 regular output
* cerr🡪 error message output
* string x
* cin >> x
* take input until a space a tab or new line.
* If you ever use a “getline” after a “cin”, use cin.ignore… after the cin and before the getline.
* A semicolon after while in the do-while loop
* X.substring (startindex,length)
* Remember to do include<cctype>
* for (int i=0; i<s1.size(); i++) {
* char lastChar=s1[i];
* }
* only declare the constant in the loop