

COSC 102

Lila Holt



C/C++ Command Line Arguments

- When executing a program in either C or C++ there is a way to pass command line arguments.
- **C:/myprogram file1.txt file2.txt**
 - myprogram -> name of program**
 - file1.txt -> command line argument 1**
 - file2.txt -> command line argument 2**
- Passed as character arrays.
- Each parameter separated by a space
- Comes into the program as two arguments
 - argc – Number of parameters (3 for example above)**
 - argv – Parameter list nameofprogram commandarg**

Command Line Arguments

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

int main(int argc, char *argv[])
{
    for (int i=0; i<argc; i++)
        cout << "This is Argument number #"
            << i
            << "->"
            << argv[i]
            << endl;
    return 0;
}
```

Sample Output

```
C:\Users\Lila\Documents\cs102\lila\commandline>cd debug
C:\Users\Lila\Documents\cs102\lila\commandline\Debug>dir
Volume in drive C is Acer
Volume Serial Number is D893-B589

Directory of C:\Users\Lila\Documents\cs102\lila\commandline\Debug
10/19/2014  04:15 PM    <DIR>          .
10/19/2014  04:15 PM    <DIR>          ..
10/19/2014  04:15 PM                5  .d
10/19/2014  04:15 PM    1,010,771  commandline.exe
10/19/2014  04:15 PM     11,809   main.cpp.o
10/19/2014  04:15 PM         28  main.cpp.o.d
           4 File(s)      1,022,613 bytes
           2 Dir(s)    399,759,474,688 bytes free

C:\Users\Lila\Documents\cs102\lila\commandline\Debug>commandline myfile1.txt myf
ile2.txt
This is how many arguments 3
This is Argument number #0->commandline
This is Argument number #1->myfile1.txt
This is Argument number #2->myfile2.txt

C:\Users\Lila\Documents\cs102\lila\commandline\Debug>commandline myfile1.txt
This is how many arguments 2
This is Argument number #0->commandline
This is Argument number #1->myfile1.txt

C:\Users\Lila\Documents\cs102\lila\commandline\Debug>_
```

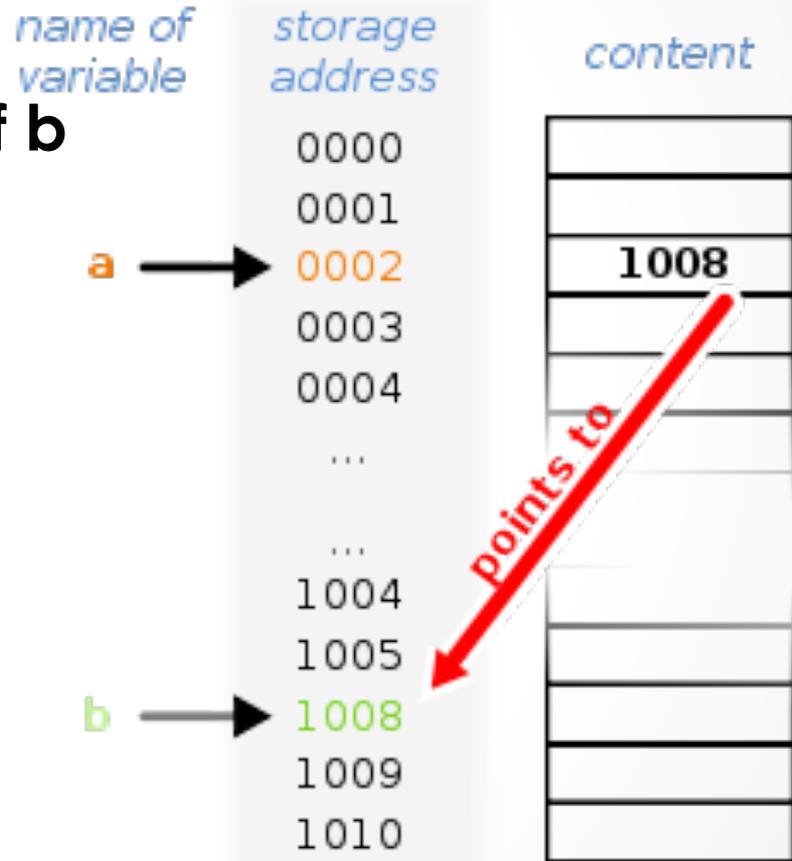
Command Line Arguments

Conventional rules:

- Arguments are always passed to `main()`.
- There must be two
 - first is an integer
 - second char pointer to an array
- First argument (`argv[0]`) will always be the name of the calling program.
- `argc` will always be at least 1
- The first argument is always `argv[0]`
- The last argument is always `argv[argc-1]`

Pointers

- Example a points to b
- a contains the address of b



Pointer

- `int main(int argc, char *argv[])`
- `char *argv[]` // the address of the character arrays
- // the * tells the computer it is a pointer

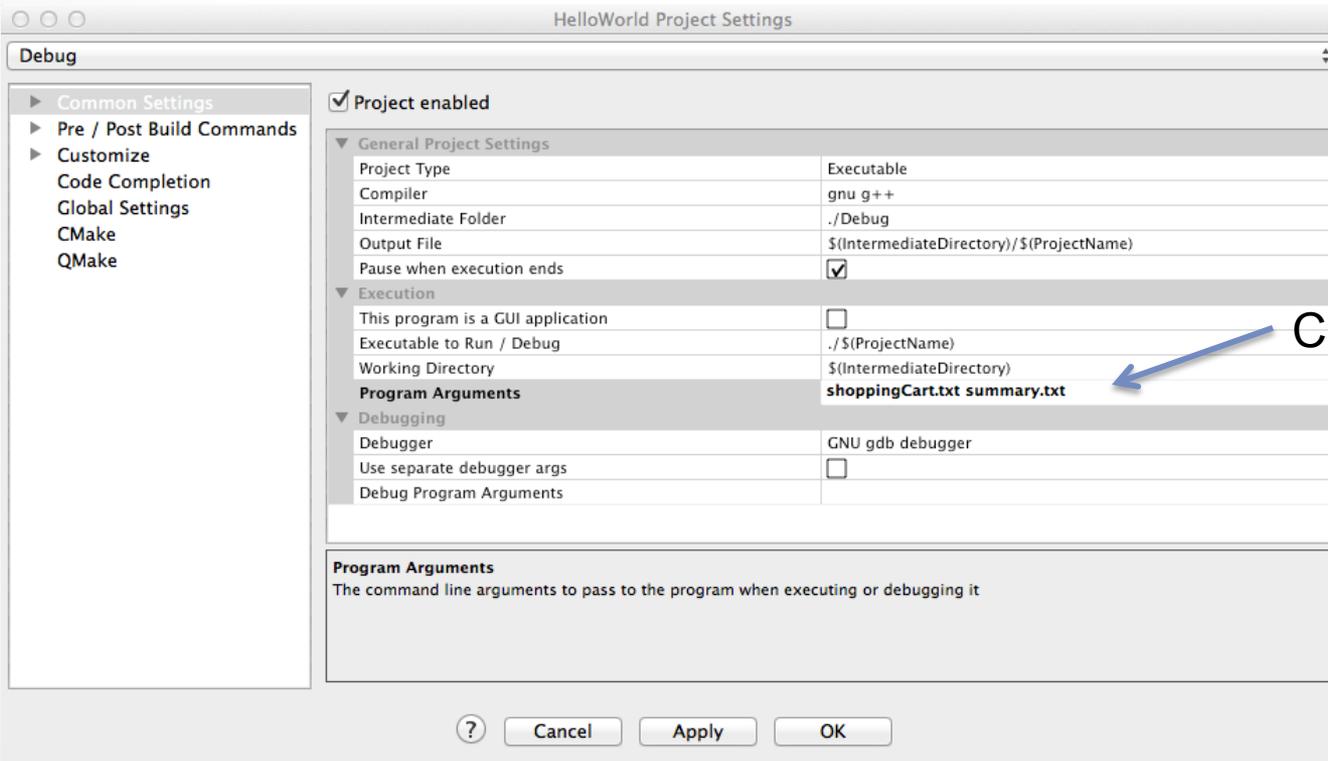
Command Line Arguments

Conventional rules:

- Arguments are always passed to `main()`.
- There must be two
 - first is an integer
 - second char pointer to an array <<< **the pointer**
- First argument (`argv[0]`) will always be the name of the calling program.
- `argc` will always be at least 1
- The first argument is always `argv[0]` << **the name of the program**
- The last argument is always `argv[argc-1]`

CodeLite – Setting Command Line Args

- **Workspace -> Open Active Project Settings**



Command Line Args