CSC 215

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Command Line Arguments

- A way of getting information into a program when it starts
- Example (command prompt)
  - I am in a directory and wish to edit a file "numbers.txt"
  - I type notepad.exe numbers.txt at the command prompt

- Example (GUI)
  - You double click a word document
  - The name of the document is then supplied as an argument to word.exe
  - word.exe opens this documents.

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Arguments to main

- Main is just a function
- Main can have parameters passed in
- The Operating System passes the arguments in when the program starts.
- There are only 2 specific arguments that are in main
  - argc – argument count – number of arguments
  - argv – argument values – array of c strings

Mains Definition - argc

- int main() will not work if we want to have parameters passed in.

  int main(int argc , char **argv)

  Argv is a 2d array of characters.

  ie. C:\> prog1.exe inputfile outputfile

  argc = 3, the name of the program, the inputfile, and the output file

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Mains Definition - argv

int main(int argc , char **argv)

argv is a 2d array of characters.

each row is a c string that represents a token

ie. C:\> prog1.exe inputfile outputfile

argv 2d arrays

p	r	o	g	1	.	e	x	e	\0	
i	n	p	u	t	f	i	l	e	\0	
o	u	t	p	u	t	f	i	l	e	\0

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Argv

- The first row is always the name of the program.
- Never modify the contents of argv, it is a special 2d array and you can crash your program.
- Always check that the user supplied the arguments before attempting to access the array

Example 1

- You have a program that needs two arguments supplied in addition to the program name, the first is the name of the input file and the second is the name of the output file.
  - ie: c:\>program.exe inputfileName outputfileName

Example 1 - continued

- Check to make sure user supplied the correct number of arguments. If not, output a usage statement (how to start program).

```
if( argc != 3 )
{
     cout << "Usage:  program.exe   inputfileName   outputfileName " << endl;
     cout << "              inputfileName – filename for the unsorted numbers" << endl;
     cout << "              outputfileName – filename to output the sorted numbers"
          << endl;
     exit(1);
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example 1 - continued

- Now that the data has been validated to exist, use it

```
fin.open( argv[1] );    // open the input file
fout.open( argv[2] );   // open the output file

if( !fin || !fout )
{
    fin.close();
    fout.close();
    cout << "Unable to open file" << endl;
    exit(1);
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example 2

- You have a program that needs two arguments supplied in addition to the program name, the first is the base name of the input file and the second is the base name of the output file.
- You need to append a .txt to each filename

ie: c:\>program.exe inputfileName outputfileName

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Example 2 - continued

- Check to make sure user supplied the correct number of arguments. If not, output a usage statement (how to start program).

```
if( argc != 3 )
{
     cout << "Usage:  program.exe   inputfileName   outputfileName " << endl;
     cout << "         inputfileName – filename for the unsorted numbers" << endl;
     cout << "         outputfileName – filename to output the sorted numbers"
          << endl;
     cout << "         both filenames will have .txt appended to the name"
          << endl;
     exit(1);
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example 2 - continued

- Now that you have validated the arguments, use them

```
strcpy ( inputName, argv[1] );   // never modify
strcat ( inputName, ".txt" );    // argv

strcpy( outputName, argv[2] );   // never modify
strcat( outputName, ".txt" );    // argv

fin.open( inputName );
fout.open( outputName);
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Listing the arguments

```
int main ( int argc, char **argv )
{
    ifstream fin;
    ofstream fout;
    int i;
    if ( argc != 3 )      //test that two arguments are present
    {
        cout << "Usage:  myprog.exe inputfile outputfile " << endl;
        cout << "Exiting now." << endl << endl;
        return -1;
    }

    for( i = 0; i < argc; i++ )   //list all the arguments, including
        cout << argv[i] << endl;  //program name
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY