File I/O and Project

How to debug your project systematically?

Not easy to debug your project

- **Issue #1**: When we see a bug, we have to **scroll up & down the console window** to find what have been done!

- **Issue #2**: Every we want to reproduce the same error and see if a change in the code fixes it, we have play the game again from the very beginning: **input many commands**!

Wish list:

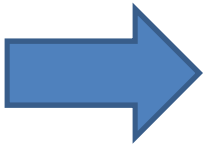
- Any method to record what I have typed?
- Can I input a list of planned commands to the game without manually typing the commands one by one again and again?

Step (1): How to debug your project?

- Record your inputs:
  - Why not use a file to record all of your inputs?

Start of program:

Open "**`steps.txt`**" in a text editor



Write down every user input in the file, e.g., **<u>the location of each move</u>**

Step (1): How to debug your project?

- Sample file (available on blackboard):
  - And you may create your own test case like this...

1 means human vs. human

alternative moves
by each player

sample1.txt - Notepad

File Edit Format View Help

```
1
43
53
64
35
46
65
56
42
25
```

Concept: I/O redirection

- A super handy way to debug and test programs

  -> This is the magic behind “code submit”

Mode	Syntax	Description
Input redirection	`[Command] < [Filename]`	Treat `[Filename]` as `stdin`
Output redirection	`[Command] > [Filename]`	Treat `[Filename]` as `stdout`
Pipe	`[Command 1] | [Command 2]`	Treat the `stdout` of `[Command 1]` as the `stdin` of `[Command 2]`

Still remember "stderr" in last lecture? It will be output redirection as "stdout"

Pipe will be used a lot in CSCI 3150 “Operating System” ☺

Concept: I/O redirection (input)

- **<u>Input redirection</u>**: don't need to type anymore!

```
C:\> basic_io.exe < haha.txt
a quick brown fox jumps ......
[ Output is the content of haha.txt ]
```

basic_io.c

```c
#include <stdio.h>

int main( void )
{
  do {
    int c = getchar() ;
    if ( c != EOF )
      putchar(c);
  } while ( c != EOF ) ;
  return 0;
}
```

```
C:\> othello.exe < sample1.txt
    1 2 3 4 5 6 7 8
 1  . . . . . . . .
 2  . . . . . . . .
 3  . . . . . . . .
 4  . . . O # . . .
 5  . . . # O . . .
......
[ The game plays automatically! ]
```

Orange Text: user input

Green Text: program output

Concept: I/O redirection (output)

- **<u>Output redirection</u>**: save your output in a file!

```
C:\> basic_io.exe > output.txt
a quick brown fox jumps
over a lazy dog
[ output.txt saves what I typed ]
```

```
C:\> othello.exe > output.txt
1
43
53
64
35
......
[ output.txt saves the game's output! ]
```

basic_io.c

```c
#include <stdio.h>

int main( void )
{
  do {
    int c = getchar() ;
    if ( c != EOF )
      putchar(c);
  } while ( c != EOF ) ;
  return 0;
}
```

Orange Text: user input

Green Text: program output

Project Testing Framework

- **We may test a program like this:**

```
C:\> othello.exe < sample_input.txt > my_output.txt
[ nothing should be printed (unless fprintf with stderr) ]

C:\>
```

What've happened?!
- " **`< sample_input.txt`** " means treating the file as the keyboard input.
- " **`> my_output.txt`** " means saving all printed characters to the file.

<u>Conclusion</u>. The game plays automatically using the sample program with
- **inputs from "`sample_input.txt`" and**
- **all its outputs go to "`my_output.txt`".**

Project Testing Framework

- "`sample_input.txt`" replaces our typing…

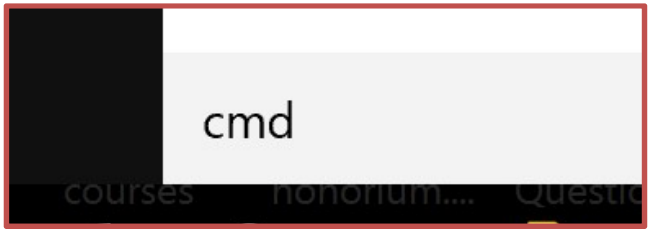
```
C:\> othello.exe < sample_input.txt
[ the game play will be printed here; you don't need to type ]

C:\> othello.exe < sample_input.txt > my_output.txt
[ nothing should be printed ]

C:\>
```

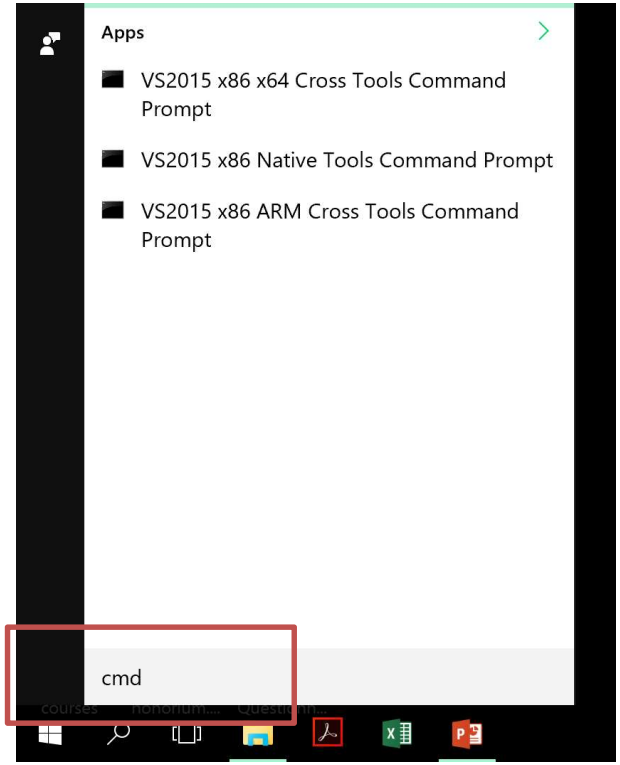
Program-Testing Framework:

- **Step (1)**. Prepare an input file. [e.g., `"sample_input.txt"`]
- **Step (2).** Start **command prompt** in Win or Mac (see next page in ppt)
- **Step (3).** Run **your program** with the input file. Optionally create an **output file** using output redirection. [e.g., `"my_output.txt"`]
- **Step (4).** Check the game board in the output file

Project Testing Framework

- How to open the command terminal in Windows?
  - Open "Start Menu" / Press the Window key
  - Type "**cmd**" and Press Enter
  - Run the first search result





Basic DOS commands (WIN)

- #1: **cd** – change the current directory (folder)

```
C:\Users\user>cd Documents

C:\Users\user\Documents>_
```

- #2: **dir** – list the contents in current directory

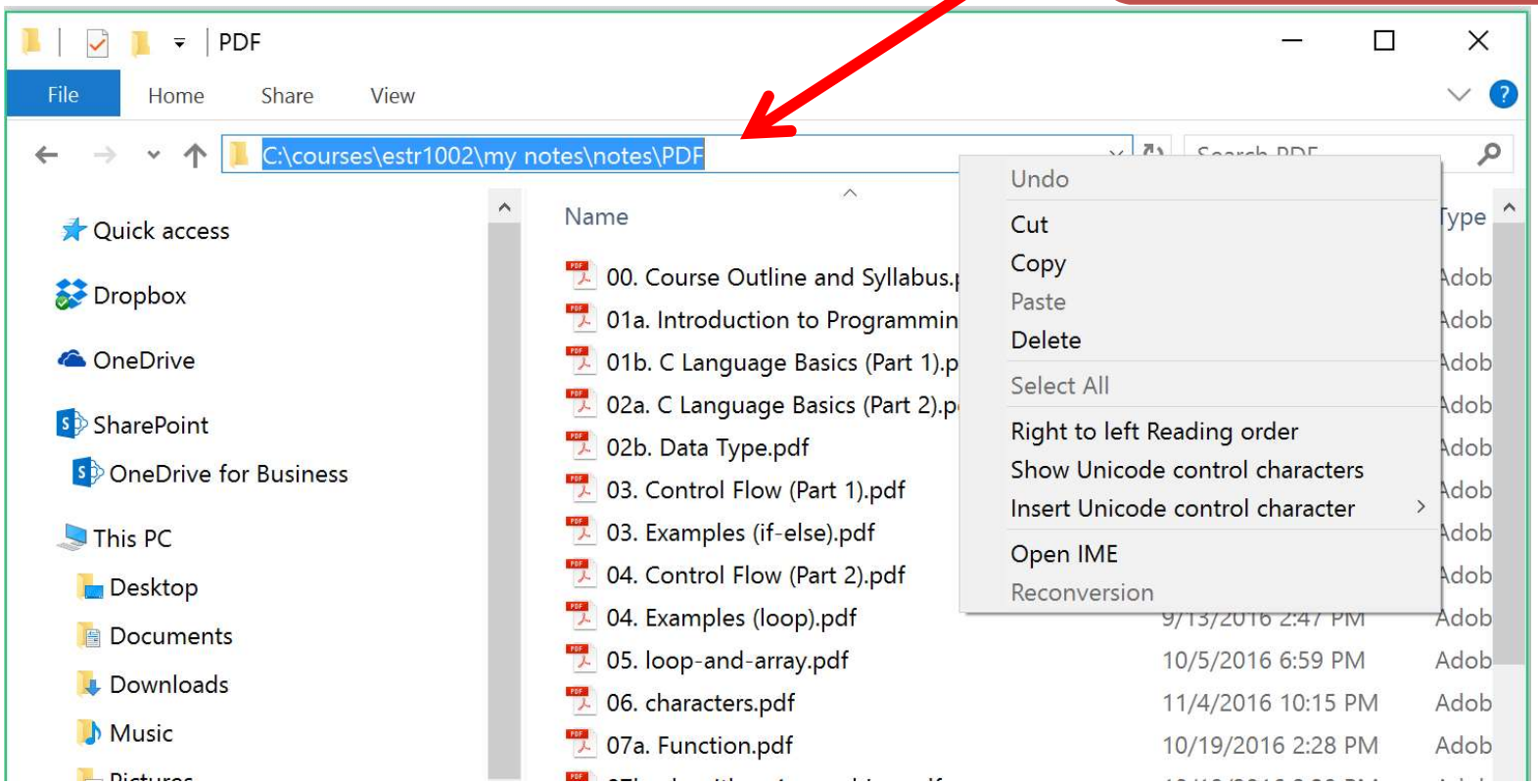
```
C:\Users\user\Documents>dir
 Volume in drive C is Home
 Volume Serial Number is EC4E-E1D9

 Directory of C:\Users\user\Documents

18/06/2014  12:11    <DIR>          .
18/06/2014  12:11    <DIR>          ..
13/11/2013  10:21    <DIR>          FinePrint files
02/06/2011  15:10    <DIR>          Visual Studio 2008
02/06/2011  15:27    <DIR>          Visual Studio 2010
               0 File(s)              0 bytes
               5 Dir(s)  18,459,672,576 bytes free

C:\Users\user\Documents>_
```

For command prompt on Mac, you have **cd** and **ls**

Project Testing Framework

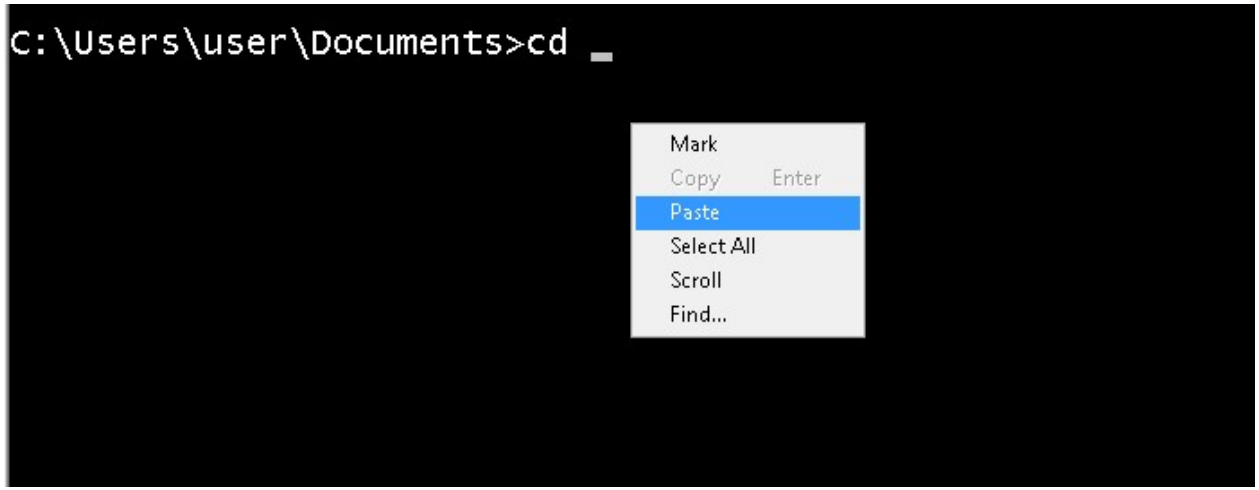
- How to reach the project folder?
  - Just use the Windows Explorer

Reach the target folder
Copy the path to clipboard, e.g., by Ctrl-C



Project Testing Framework

- Type "**cd**" and a space on the terminal
- Right-click on the terminal and select "**Paste**"



- You can always ask the lecturer and TA how ☺