WESTERN SYDNEY
UNIVERSITY

Basic Input / Output (I/O)

ELEC1006: ENGINEERING COMPUTING

Standard Output (`cout`)

WESTERN SYDNEY UNIVERSITY

- Displays output on the computer screen.
- Requires `iostream` header file.
- You use the stream insertion operator `<<` to send output to `cout`:

```
cout << "Programming is fun!";
```

- Can be used to send more than one item to cout:

```
cout << "Hello " << "there!";
```

Or:

```
cout << "Hello ";
cout << "there!”;
```

Displaying with `cout`

WESTERN SYDNEY UNIVERSITY

- This produces one line of output:

```
cout << "Programming is ";
cout << "fun!";
```

- You see on screen –

```
Programming is fun!
```

- You can use the **`endl`** manipulator to start a new line of output. This will produce two lines of output:

```
cout << "Programming is" <<
endl;
cout << "fun!”
```

- You see on screen –

```
Programming is
fun!
```

New line with `cout`

WESTERN SYDNEY UNIVERSITY

- You do NOT put quotation marks around **`endl`**
- The last character in **`endl`** is a lowercase L, not the number 1.
- You can also use the **`\n`** escape sequence to start a new line of output. This will produce two lines of output:

```
cout << "Programming is\n";
cout << "fun!";
```

- You see on screen –

```
Programming is
fun!
```

Standard Input (`cin`)

WESTERN SYDNEY UNIVERSITY

- Standard input object.
- Like `cout`, requires `iostream` header file.
- Used to read input from keyboard
- Information retrieved from `cin` with `>>`
- Input is stored in one or more variables

Acquiring with `cin`

WESTERN SYDNEY UNIVERSITY

- Input one value:

```
int height;
cout << "How high is the ceiling? ";
cin >> height;
```

- Can be used to input more than one value:

```
int height, width;
cout << "How high is the ceiling? ";
cin >> height >> width;
```

- Multiple values from keyboard must be separated by spaces
- Order is important: first value entered goes to first variable, etc.

Reading Strings with `cin`

WESTERN SYDNEY UNIVERSITY

- Can be used to read in a string
- Must first declare an array to hold characters in string:

```
char myName[21];
```

- `myName` is the name of the array, `21` is the number of characters that can be stored (the size of the array), including the NULL character at the end
- Can be used with `cin` to assign a value:

```
cin >> myName;
```

- `cin` extraction stops reading as soon as it finds any blank space character.

`cin` and Strings

WESTERN SYDNEY UNIVERSITY

- If the input data type is declared as string, `cin` could be used to extract the entire input line including blank spaces.

```
// cin with strings
#include <iostream>
#include <string>
using namespace std;

int main ()
{
  string mystr;
  cout << "What's your name? ";
  getline (cin, mystr);
  cout << "Hello " << mystr << ".\n";
  cout << "What is your favorite team? ";
  getline (cin, mystr);
  cout << "I like " << mystr << " too!\n";
  return 0;
}
```

From cplusplus.com

Read in the string

More info on C++

WESTERN SYDNEY UNIVERSITY

[1] cplusplus.com: Basic Input/Output
https://cplusplus.com/doc/tutorial/basic_io/