

# CSCI 1370

FEBRUARY 13, 2016

ADMINISTRATIVA

# QUARTER EXAM #1 (10pts)

## STATISTICS (excluding extra credit/bonuses)

- Range: 4.70–9.50
- Median: 8.25

## NOTES

- Group bonuses still to be calculated.
- Exams will be returned next Monday.
- Better performance on later exams replace prior scores.

## LABS/EXERCISES (13.5pts)

- Submissions will be accepted until the last day of class (May 3).
- Grading of these items is thus postponed (to May 4).
- Grades will be computed as the percent (e.g., of labs) completed multiplied by the respective number of points (e.g., 6.5).

# MISCELLANEOUS

## QUIZZES

- ID switched to name (to save TA time)

## EXTRA CREDIT

- Attend Professor Dietrich's problem-solving seminars (Thursdays, 12:15-1:15pm): +.25pts per seminar.

## CONNECTING W/ADDITIONAL PEERS

- Search for/join the Facebook group (UTRGV Computer Science 1).

REVIEW:

WRITING & USING FUNCTIONS

# GROUP EXERCISE

Write a program that prints "Hello world!" in the console.

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

int main()
{
    cout << "Hello world!";
}
```



# GROUP EXERCISE

Write a program that prints "Hello world!" in the console. Now add a function to the program that prints "Goodbye!" in the console.

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

void sayGoodbye()
{
    cout << "Goodbye!";
}

int main()
{
    cout << "Hello world!";
}
```

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

```
void sayGoodbye()
```

```
{
    cout << "Goodbye!";
}
```

```
int main()
{
    cout << "Hello world!";
}
```

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

```
void sayGoodbye()
{
    cout << "Goodbye!";
}
```

```
int main()
{
    cout << "Hello world!";
}
```

The **return type**. It tells the computer what kind of data (if any) is being *passed* back to the caller.

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

```
void sayGoodbye()
{
    cout << "Goodbye!";
}
```

```
int main()
{
    cout << "Hello world!";
}
```

The function **name**.

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

void sayGoodbye()  
{  
    cout << "Goodbye!";  
}  
  
int main()  
{  
    cout << "Hello world!";  
}
```

The **parameters list**. It tells the computer what kind of data (if any) is *sent* from the caller.

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

```
void sayGoodbye()
```

```
{
    cout << "Goodbye!";
}
```

```
int main()
```

```
{
    cout << "Hello world!";
}
```

# GROUP EXERCISE

Write a program that prints "Hello world!" in the console. Now add a function to the program that prints "Goodbye!" in the console.

How can we modify our program to print both ("Hello world!" and "Goodbye!") in the console?



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

void sayGoodbye()
{
    cout << "Goodbye!";
}

int main()
{
    cout << "Hello world!";
    sayGoodbye();
}
```

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

void sayGoodbye()
{
    cout << "Goodbye!";
}

int main()
{
    cout << "Hello world!";
    sayGoodbye();
}
```

This is a function **call**. It tells the computer to execute whatever code is in a function named `sayGoodbye`.

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

```
void sayGoodbye()
{
    cout << "Goodbye!";
}
```

```
int main()
{
    cout << "Hello world!";
    sayGoodbye();
}
```

FUNCTIONS (CONT.):

PASSING DATA TO AND FROM FUNCTIONS

# FUNCTION: SUM

```
int sumOf(int a, int b)
{
    return a + b;
}
```

# FUNCTION: SUM

```
int sumOf(int a, int b)
{
    return a + b;
}
```

# FUNCTION: SUM

```
int sumOf(int a, int b)
{
    return a + b;
}
```

# FUNCTION: SUM

```
int sumOf(int a, int b)
{
    return a + b;
}
```



# FUNCTION: SUM

```
int sumOf(int a, int b)
{
    return a + b;
}
```

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

```
int sumOf(int a, int b)  
{  
    return a + b;  
}
```

```
int main()  
{  
    int result = sumOf(5, 10);  
    cout << sumOf(5, 10);  
}
```

FUNCTIONS (CONT.):  
THE **#INCLUDE** DIRECTIVE & USING PRE-  
DEFINED FUNCTIONS

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

```
int main()
{
    double x = 4;

    cout << x;
}
```

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

```
int main()
{
    double x = 4;

    cout << x;
}
```

This is the **#include directive**. It directs the compiler to *include* (prepend) the **iostream library** in this program.

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

```
int main()
{
    double x = 4;
```

```
    cout << x;
```

```
}
```

Including the `iostream` library allows us to use things like `cin` and `cout`.

A C++ **library** is a set of functions, operators, and types – already written and packaged for others to use. For example, the `iostream` library allows us to do things like print to the console (e.g., `cout << "hello"`) and save console input (e.g., `cin >> x`).

# USING LIBRARIES: `iostream`

The insertion operator (`<<`) is **built-in**. However, the computer needs to know *where* to print...

The `iostream` library:

- Defines the type `ostream` (output stream). Output streams move characters, one by one and in a fixed order, from the program to an output location (the screen, a file, etc.).
- Declares the variable `cout` (`cout` is of type `ostream` and is programmed to print to the console).



# USING LIBRARIES: `iostream`

The extraction operator (`>>`) is also **built-in**. However, the computer needs to know *from where* to extract information...

The `iostream` library:

- Defines the type `istream` (input stream). Input streams move characters, one by one and in a fixed order, from a specified location to the program.
- Declares the variable `cin` (`cin` is of type `istream` and is programmed to extract information from the console).

# THE `#include` DIRECTIVE

To use a library, you must tell the program to include its **header file**.

The syntax for including the `iostream` library is:

```
#include <iostream>
```

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

```
int main()
{
    double x = 4;

    cout << x;
}
```

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

int main()
{
    double x = 4;
    double result = sqrt(x);
    cout << x;
}
```

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

int main()
{
    double x = 4;
    double result = sqrt(x);
    cout << x;
}
```

Another **#include directive**.  
It directs the compiler to include the **math library** here.

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

int main()
{
    double x = 4;
    double result = sqrt(x);
    cout << x;
}
```

As a result, we can use various **pre-defined math functions.**

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

```
int main()
{
    double x = 4;
    double result = sqrt(x);
    cout << x;
}
```

# SOME `cmath` LIBRARY FUNCTIONS...

- `sqrt(x)`:
  - Requires one input parameter (**x**) of type **double**.
  - Returns a **double** (the square root of **x**).
- `pow(x, y)`:
  - Requires two input parameters (**x**, **y**) of type **double**.
  - Returns a **double** ( $x^y$ ).



# SOME `cmath` LIBRARY FUNCTIONS...

- `floor(x)`:
  - Requires one parameter (**x**) of type **double**.
  - Returns a **double** (the largest whole number  $\leq x$ ).
- `ceil(x)`:
  - Requires one input parameter (**x**) of type **double**.
  - Returns a **double** (the smallest whole number  $\geq x$ ).

# EXERCISE

**With a partner,** write out the function **headers** for the `floor` and `ceil` (predefined) math functions.

# SOME **cctype** (CHARACTER) LIBRARY FUNCTIONS...

- **tolower(x)**:
  - Requires one parameter (**x**) of type **int** or **char**.
  - Returns an **int** (the lowercase value of **x**).
- **toupper(x)**:
  - Requires one parameter (**x**) of type **int** or **char**.
  - Returns a **double** (the uppercase value of **x**).

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

```
int main()
{
    char c, lowercase, uppercase;
    cout << "Enter a letter:";
    cin >> c;
    lowercase = tolower(c);
    uppercase = toupper(c);
}
```

# EXERCISE

**With a partner**, write out the function **headers** for the `tolower` and `toupper` (predefined) character functions.