

# FUNCTIONS, LOOPS

---

Problem Solving with Computers-I

C++

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

int main(){
    cout<<"Hola Facebook!n";
    return 0;
}
```

GitHub



# Pass by value

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

void bar(int x){
    x = x + 5;
}

int main(){
    int y = 0;
    bar(y);
    cout<<y;
    return 0;
}
```

What is printed by this code?

- A. 0
- B. 5
- C. Something else

## The accumulator pattern

Write a function that takes a parameter  $n$  and prints the sum of the series:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Write another function that returns the sum of the series

# Formatting output to terminal

See pages 91 and 190 of textbook

```
int i =10;
double j = 1/static_cast<double>(i);
cout.setf(ios::fixed);      // Using a fixed point representation
cout.setf(ios::showpoint); //Show the decimal point
cout.precision(3);
cout<<j;
```

What is printed by the above code?

- A. 0
- B. 0.1
- C. 0.10
- D. 0.100
- E. None of the above

# Nested for loops – ASCII art!

Write a function that prints a square of a given width

```
drawSquare(5);
```

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

# Draw a triangle

Which line of the drawSquare code  
(show on the right) would you modify  
to draw a right angled triangle

```
drawTriangle(5);
```

```
*  
* *  
* * *  
* * * *  
* * * * *
```

```
6   for(int i = 0; i < n; i++){ //A  
7       for(int j=0; j < n; j++){ //B  
8           cout<<"* "; //C  
9       }  
10      cout<<endl; //D  
11  }  
12  cout<<endl; //E  
13
```

# Infinite loops

```
for(int y=0;y<10;y--)  
    cout<<"Print forever\n";
```

```
int y=0;  
for(;;y++)  
    cout<<"Print forever\n";
```

```
int y=0;  
for(;y<10;);  
    y++;
```

```
int y=0;  
while(y<10)  
    cout<<"Print forever\n";
```

```
int y=0;  
while(y=2)  
    y++;
```

# PROGRAM I/O

- **What are two ways for a user to provide input to a C++ program**



# Passing arguments to main (via the command line)

- We can pass information into a C++ program through the command line when executing the program.
- The main function will need to have the following:

```
int main(int argc, char* argv[ ])
```

- `int argc` is the number of "arguments" the program has, including the executable name.
- `char* argv[ ]` is the "list" of arguments passed into the program.
  - `argv[0]`: name of the program
  - `argv[1]`: 1<sup>st</sup> argument, remember this is a C-string
  - Use `atoi` to convert a C-string to a number `atoi(argv[1])`

# Next time

- Automating the compilation process with Makefiles
- Intro to lab02