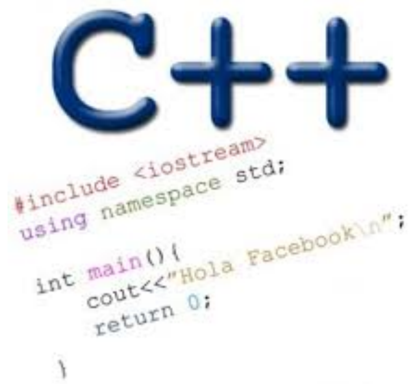
LOOPS (REVIEW)
C/C++ MEMORY MODEL
FUNCTIONS
VARIABLE SCOPE AND LIFETIME

Problem Solving with Computers-I



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

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

Clickers out – frequency AC

C++ types in expressions

```
int i =10;

double sum = 1/i;

cout<<sum;
```

What is printed by the above code?

A. 0

B. 0.1

C. 1

D. None of the above

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 the output?

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

C++ for loops

For loop is used to repeat code (usually a fixed number of times)

General syntax of a for loop:

```
for (INITIALIZATION; BOOLEAN_EXPRESSION; UPDATE) {
        // code to repeat
}
```

The accumulator pattern

Write a program that calculates the series:
1+ 1/2+ 1/3+ ….1/n, where `n` is specified by the user

Nested for loops – ASCII art!

Write a program that draws 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++;
```

Functions: Basic abstraction in programs

- Keep programs DRY !
- Three steps when using functions
  1. DECLARE: void drawSquare(int y);

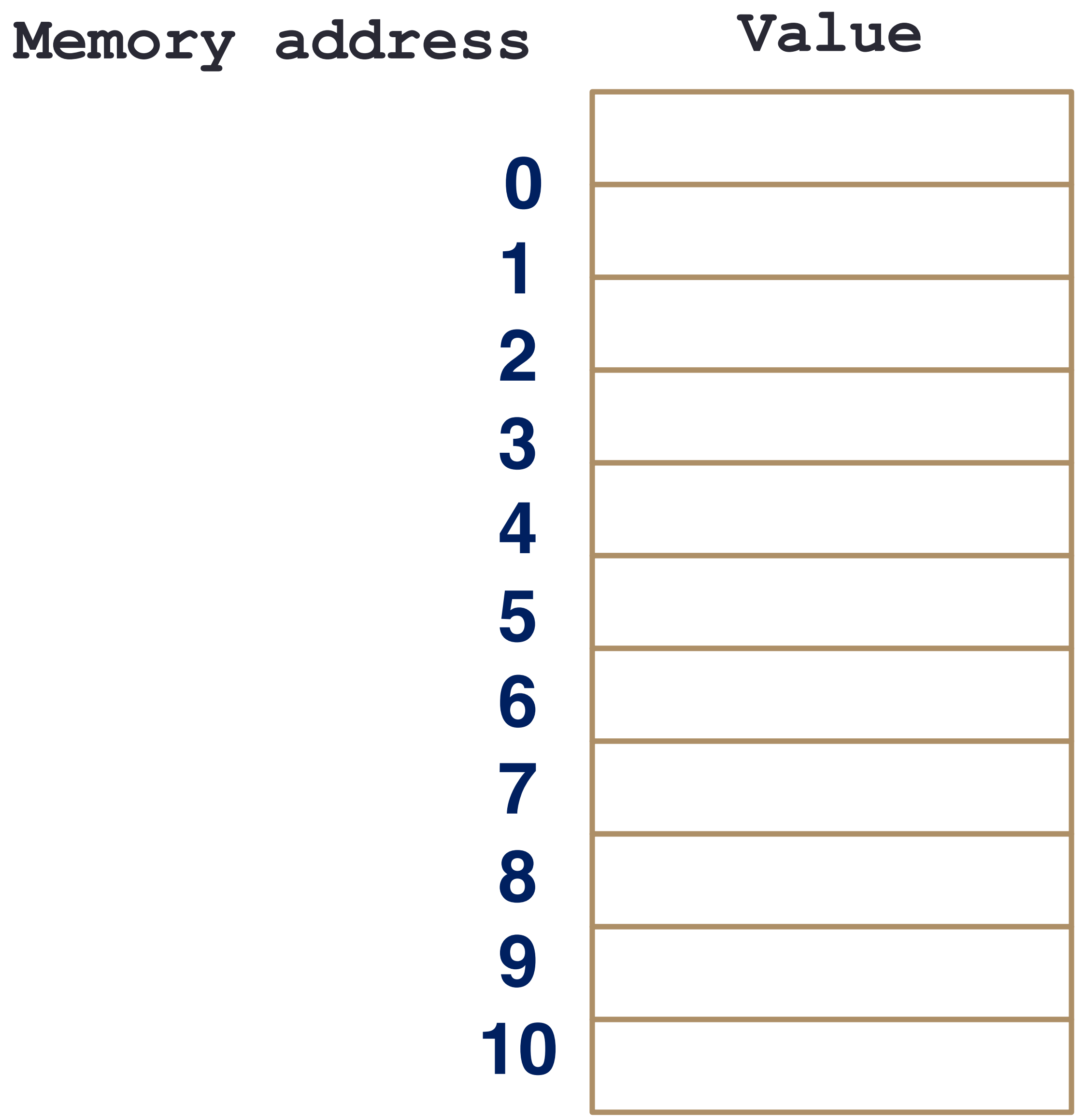
  2. DEFINE: Write the actual code inside the function

  3. CALL: drawSquare(20);

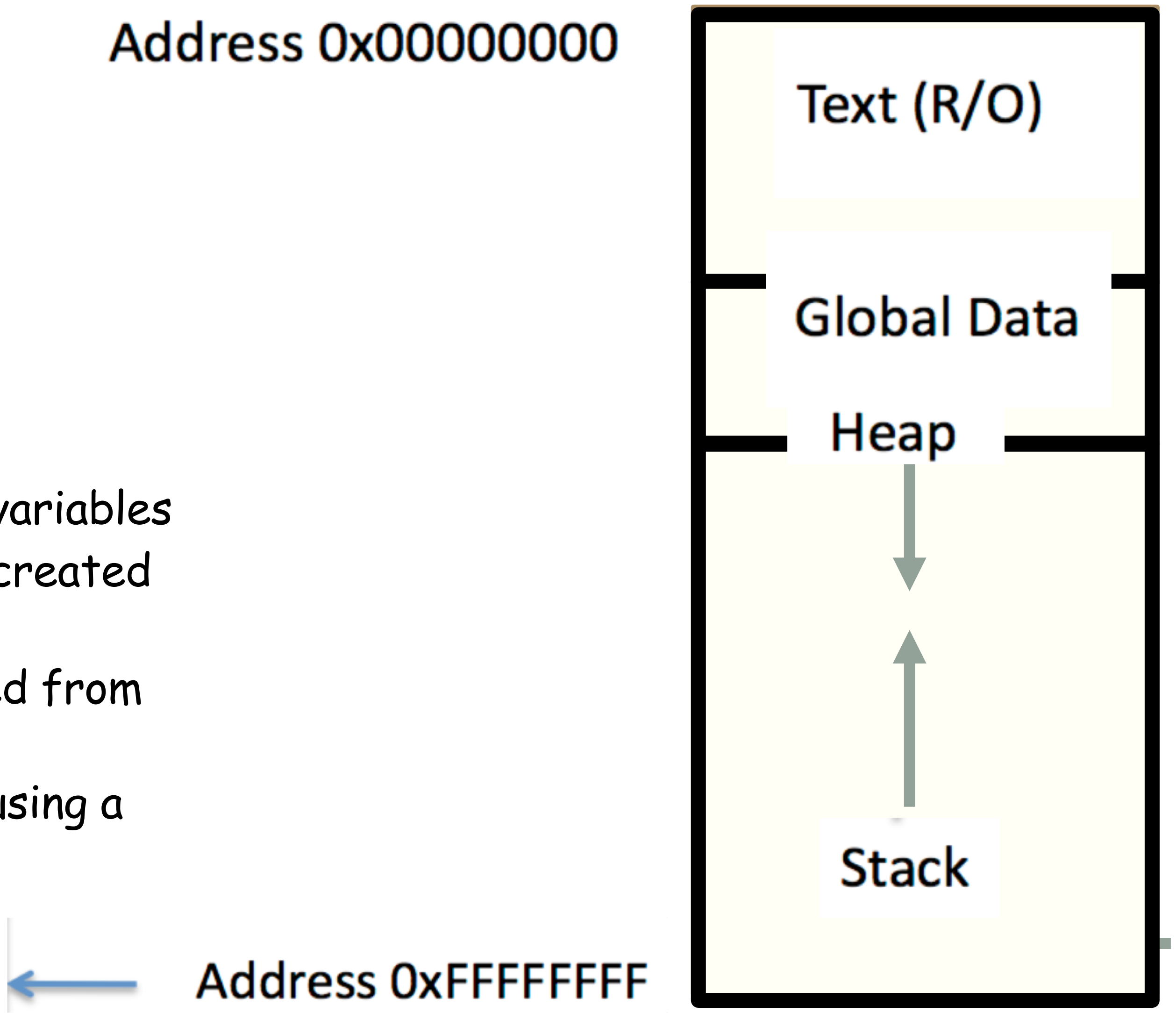
You must always declare/define functions before calling them.
Demo the use of functions

General model of memory

- Sequence of adjacent cells
- Each cell has 1-byte stored in it
- Each cell has an address (memory location)



C++ Memory Model



Stack: A region in program memory to “manage” local variables

Every time a function is called, its local variables are created on the stack

When the function returns, local variables are removed from the stack

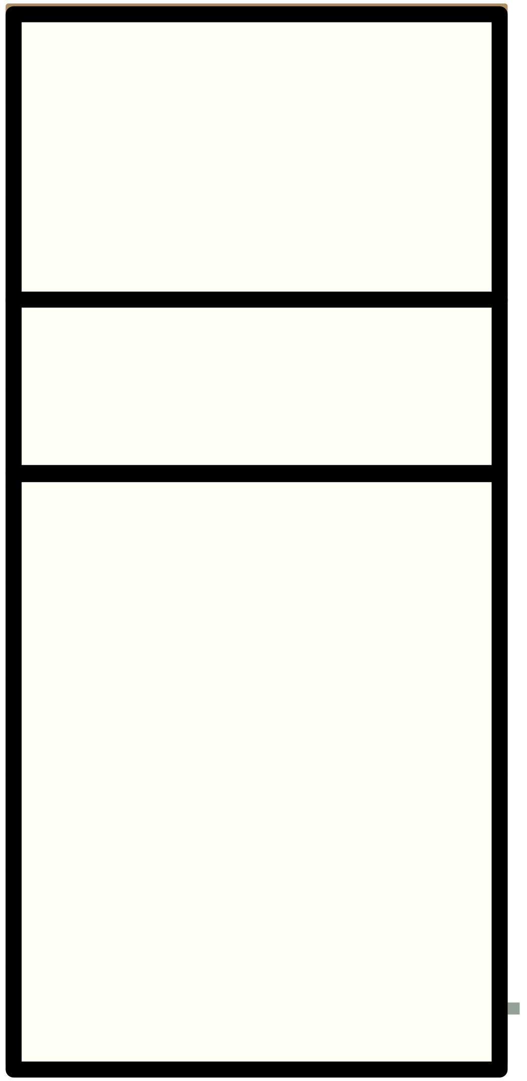
Local variables are created and deleted on the stack using a Last in First Out principle

Variable: scope: Local vs global

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

int b_out;

int bar(){
      b_out = 20;
      int a_in = b_out+5;
      return a_in;

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

Which variables are in memory when program execution reaches the line:
 cout<<result;

A. result
B. b_out
C. a_in
D. A and B
E. None of the above



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

Next time

- Automating the compilation process with Makefiles
- Intro to lab02