

# Draw a triangle

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

```
./drawTriangle
Enter the length of the base
5

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

```
5 int main(){
6     int side;
7     cout<<"Enter the length of the base"<<endl;
8     cin>>side;
9
10    for(int j = 0; j < side; j++){ //A
11        for(int i=0; i < side; i++){ //B
12            cout<<"*"; //C
13        }
14        cout<<endl; //D
15    }
16    cout<<endl; //E
17
18 }
```

# 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, MORE LOOPS

---

Problem Solving with Computers-I

<https://ucsb-cs16-sp17.github.io/>

C++

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

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



Clickers out – frequency AB

# Functions: Basic abstraction in programs

- Functions keep you DRY!
- Three steps when using functions
  1. DECLARE
  2. DEFINE
  3. CALL

# What is the value of argc in each of these cases?

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

```
$/drawTriangle
```

```
$/drawTriangle 5
```

```
$/drawTriangle 5 cat dog fizz
```

# Passing parameters to programs

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

```
$/drawTriangle
```

```
Enter the length of the base  
5
```

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

```
$/drawTriangle 5
```

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

# Control Flow: while and do while loops

```
while(Boolean expression) {  
    //statement 1  
    //statement 2  
}
```

```
do {  
    //statement 1  
    //statement 2  
}while(Boolean expression);
```

Identify the code that is not equivalent to the other two?  
Assume 'n' is an integer that has already been declared (may be positive or negative)

**A.**

```
for( int x = 0; x < n; x++ ) {  
    cout<<x <<endl;  
}
```

**B.**

```
int x = 0;  
while(x < n) {  
    cout<< x << endl;  
    x++;  
}
```

**C.**

```
int x = 0;  
do{  
    cout<< x<< endl;  
    x++;  
} while(x < n);
```

**D.** They are ALL equivalent



Practice: Use while loops to print a sequence:  $x_{\min}$ ,  $x_{\min}+1$ ,  $x_{\min}+2$ , ...,  $x_{\max}$  for user specified inputs  $x_{\min}$  and  $x_{\max}$

Sample run of the program: (You must use while loops, and specify the limits of the sequence as inputs to your program)

```
$ ./print_series 10 15  
10, 11, 12, 13, 14, 15
```

# Function call mechanics

What is the output of the following code

```
int sum(int a, int b){  
    return a+b;  
}  
int main(){  
    int result =0;  
    int x =10, y =20;  
    result = sum(x, y);  
    cout<<result+30;  
}
```

# Function call mechanics

What is the output of the following code

```
int sum(int a, int b){
    cout<< a+b;
}
int main(){
    int result =0;
    int x =10, y =20;
    result = sum(x, y);
    cout<<result+30;
}
```

# Function call mechanics

What is the output of the following code

```
int sum(int a, int b){
    int result= a+b;
    exit(0);
}

int main(){
    int result =0;
    int x =10, y =20;
    result = sum(x, y);
    cout<<result+30;
}
```