Today’s content

- Step 1 & 2: revise the Tic-Tac-Toe game (remove global variables, declare the variables in main instead and pass them to functions as needed).

- Step 3: Implement moveIsValid() function that checks if an intended move is valid.

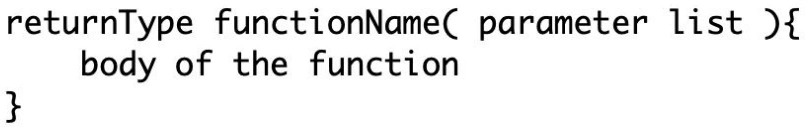
Minimize the use of global variables

- Why?
  Global variables can be altered by any part of the code, making it difficult to maintain the program (see *Zybooks 5.9 Scope of variable* for more details).

- Global variables are typically limited to const variables such as π (3.1415).

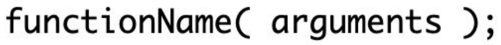
Questions

- How to declare a variable?
- How to define a function?
  A function definition looks like this:



```
returnType functionName( parameter list ){
    body of the function
}
```

- How to make a function call?
  A function call looks like this:



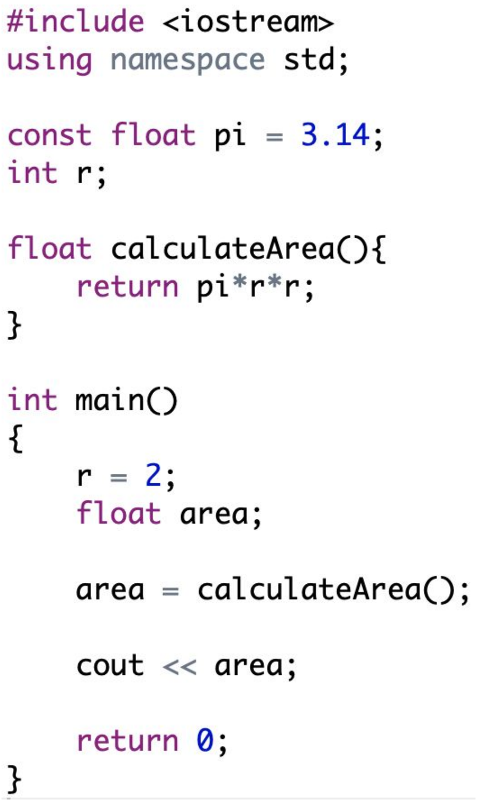
```
functionName( arguments );
```

- When/ how to use call-by-reference?

Example: use a function to calculate the area of a circle

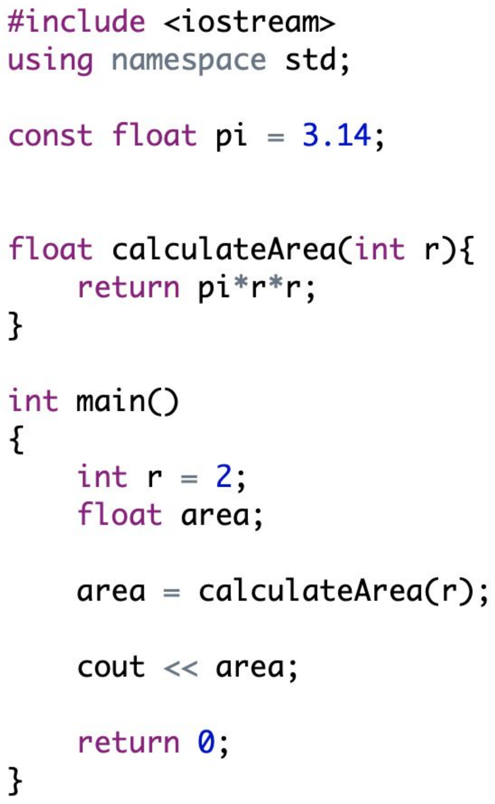
area = pi*r*r, where pi = 3.14 and r is the radius of the circle.

Version 1:
r is a global variable



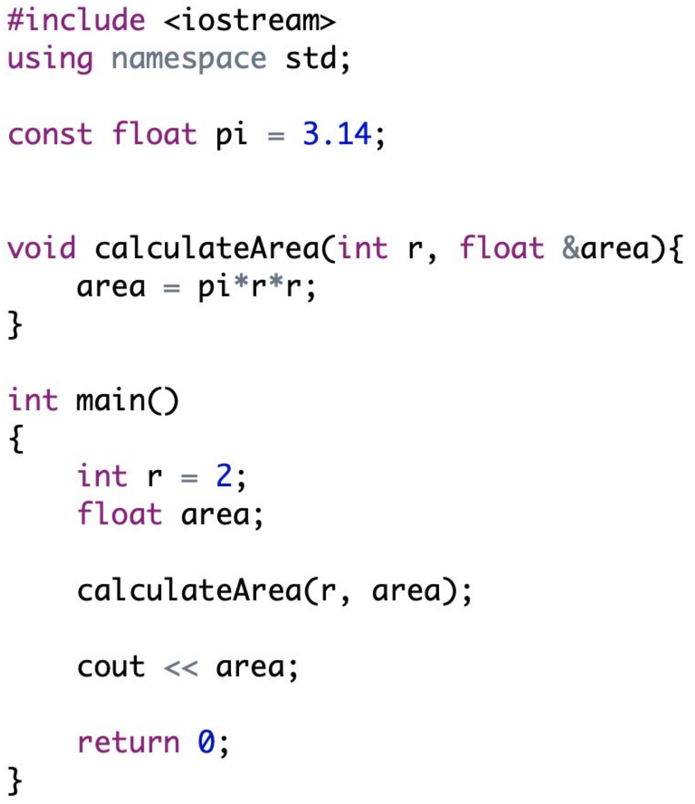
```cpp
#include <iostream>
using namespace std;

const float pi = 3.14;
int r;

float calculateArea(){
    return pi*r*r;
}

int main()
{
    r = 2;
    float area;

    area = calculateArea();

    cout << area;

    return 0;
}
```

Version 2:
r is a local variable of main and is passed to calculateArea()



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

const float pi = 3.14;


float calculateArea(int r){
    return pi*r*r;
}

int main()
{
    int r = 2;
    float area;

    area = calculateArea(r);

    cout << area;

    return 0;
}
```

Version 3:
area is a local variable of main and is passed to calculateArea() using call by reference



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

const float pi = 3.14;


void calculateArea(int r, float &area){
    area = pi*r*r;
}

int main()
{
    int r = 2;
    float area;

    calculateArea(r, area);

    cout << area;

    return 0;
}
```