



```
halloween--;  
while(classInSession == true)  
{  
    payAttention();  
    //if(basketTurn == true)  
    //  takeChocolate();  
}
```

Review – Building a function

1. Function Prototype

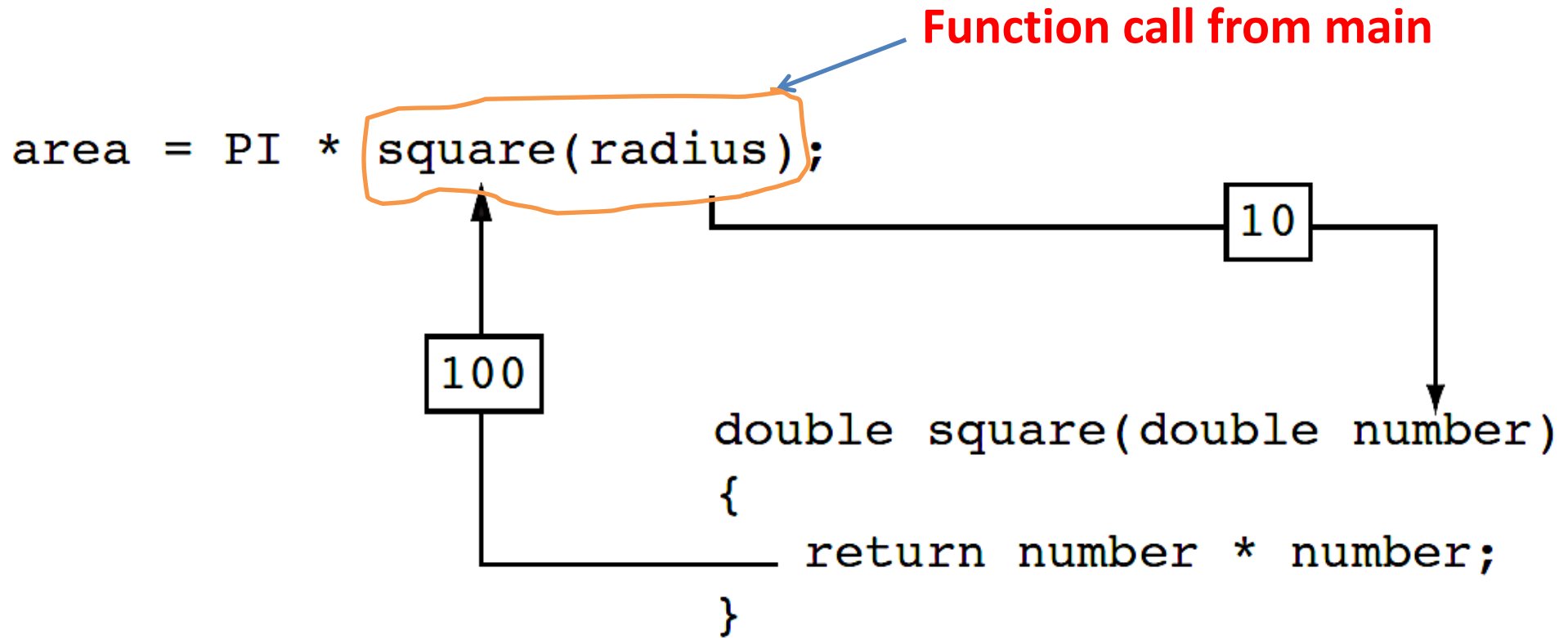
```
returnType functionName (parameter-list);  
void evenOrOdd(int);
```

2. Function Definition

```
returnType functionName (parameter-list)  
{  
    function statments;  
}
```

- To return a value, use the keyword **return**, followed by an expression that matches the expected return type
 - *return expression;*
- **A function can take multiple values but can only "return" one value**

Another Example



- If you assign the return value of the square function to a variable, the variable must be a double.
- If the variable is an int, the value will be truncated

Returning a Value From a Function

Concept

- The prototype and the definition **must** indicate the data type of return value

```
int sum(int num1, int num2)
```

- **How the calling function handles the returned value:**

1. assign it to a variable
2. send it to `cout`
3. use it in an expression (prev. slide)

Exercise: write an example for each using `sum` function

```
int  x = 10, y = 15;  
cout<<"The sum is " << sum(x, y) <<endl;
```

How does this work?

Recall

```
2          // the value of this expression is 2
2 + 3      // the value of this expression is 5
2 > 3      // the value of this expression is 0 (false)
i = 2      // the value of this expression is 2
```

For a value returning functions:

- Function calls **are expressions** that are evaluated to a value

```
total = sum(6, 9);  //the value of sum(6,9) is 15
```

//or use it in an expression

```
total = 5 + sum(6, 9);  //total = 20 after this statment
```

//or send it to cout

```
cout << sum(6, 9);
```

6.9 Returning a Boolean Value

- Function can return `true` or `false`
- Declare return type in function prototype and heading as `bool`
- Function body must contain `return` statement(s) that return `true` or `false`
- **The calling function can use return value in a relational expression**

Feedback Quiz

Write the **function** (`isValid`) that **accepts (receives)** an `int` argument and **returns** `true` if the argument is within the range of 1 through 100, or `false` otherwise

```
bool isValid(int num)
{
    if (num >= 1 && num <= 100)
        return true;
    else
        return false;
}
```

OR

```
bool isValid(int num)
{
    bool status;
    if (num >= 1 && num <= 100)
        status = true;
    else
        status = false;
    return status;
}
```