WEEK SIX

Acknowledgements: Slides created based off material provided by Dr. Travis Doom

THE METHOD

- Smaller, simpler, subcomponent of a program
- Hides low-level details, making program easier to understand
- Helps promote efficient coding and limit unnecessary repetition
- Methods must be declared/defined
- AKA functions, procedures, subroutines

THE METHOD PARTS

```
public static int sum (int num1, int num2) Method Header
public static int sum (int num1, int num2) Definition
{
        int result = num1 + num2;
        return result;
}
totalGrade = sum(grade1, grade2); Method call
```

METHOD HEADER (DECLARATION)

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

Access Modifier
- defines where method can be accessed or called from

Static Modifier
- Use this when we don't need an object for the method to run
- Ex: methods in our main class

Return Type
- data type of the variable this method will return
- if it doesn't return anything, we use the keyword 'void'

Method Name
- camel case
- can be whatever we want
- describe method's purpose

data type for 1st param

name for 1st param

data type for 2nd param

name for 2nd param

DEFINITION (METHOD HEADER & BODY)

```
public static int sum (int num1, int num2)] method header

{

        int result = num1 + num2; // method code: takes two integers from the
                                     user and adds them together

        return result;

}
```

- return keyword is used to indicate to the compiler that we want it to go back to the section of code the method was called from
- it also allows us to pass a value back to the main code just like parameters allow us to pass values into the method

* data type of result must match return type in the method header

METHOD CALL

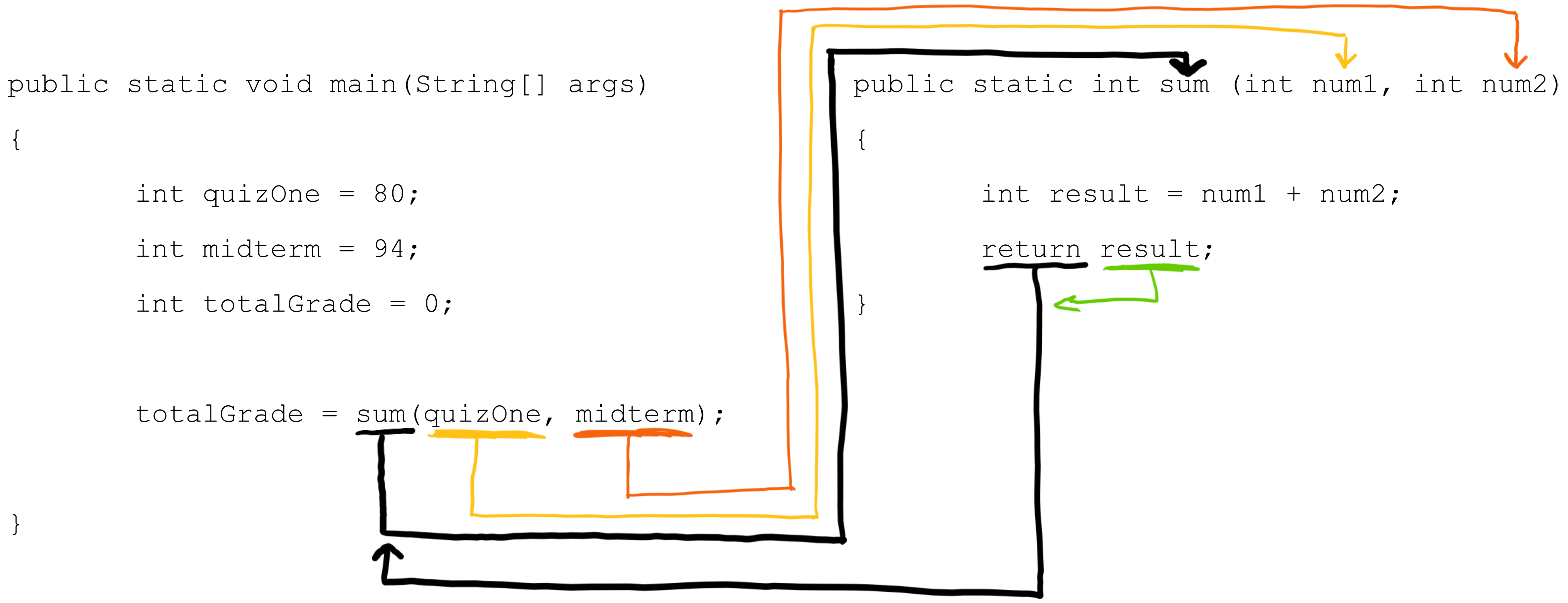
- method call will evaluate to whatever value is being returned
- in this case, the value in result would get stored in totalGrade

```
int totalGrade = sum(grade1, grade2);
```

data type of totalGrade must match the return type of the method

We don't have to specify a class/object for the method if we are in the same class as the method

WHAT HAPPENS WHEN WE CALL A METHOD



```
public static void main(String[] args)
{
        int quizOne = 80;
        int midterm = 94;
        int totalGrade = 0;

        totalGrade = sum(quizOne, midterm);

}
```

```
public static int sum (int num1, int num2)
{
        int result = num1 + num2;
        return result;
}
```

NOTES ON METHOD PARTS

- A method is identified not only by its name but also by the parameters it takes in
    - `public static int abs(int a)`
    - `public static long abs(long a)`
    - `public static float abs(float a)`
- Parameter order and data type matter
- If the method doesn't return anything, the return type is `void`
    - E.g. `System.out.println("Hi!");`
- If the method does return something, it must match the return type

SCOPE NOTES

- Access modifier for method defines where the method can be called from
- Variables declared within a method only exist within that scope (not accessible outside method)
- Parameters passed into a method become local to that method
  - Pass by value
  - Changes to the local copies of the variables do not change the original
  - Must utilize return to make changes to primitive data types
- You can pass a reference to an object to a method
  - Then the method can modify that object
  - Pass by reference
  - Don’t have to return something to make changes to the parameter

IN CLASS ACTIVITY

- Write a method that calculates and returns the average of three grades
- First, write the method header
- Then, write the method body
- Finally, write code to call the method from the main method and print out the result

TRICK OR TREAT ACTIVITY

- Write a program that prompts the user to type trick or treat
  - If the user types trick, pick a random scary message to print out (like “BOO!” or “I want brains!”)
  - If the user types treat, pick a random candy name to print out (like “Snickers” or “Kit-Kat”)
  - For both options, there must be at least 3 possible messages that can print out
- At the beginning, ask the user how many houses they would like to visit
  - This is the number of times you will prompt the user for a trick or treat
- Your solution should utilize:
  - Method(s)
  - Loop(s)
  - Conditional Statement(s)

METHOD REVIEW

- Write a method that checks if a person meets the requirements to attend a specific institution
  - To attend, their GPA must be at least 2.7
  - They must have an ACT score of at least 25
  - They must have submitted a college essay
- Determine the correct data types for both the return type and parameters
- Write the logic to check if the provided parameters meet the requirements

METHOD REVIEW

- Write a method that takes in a value to represent how high we want to count and another value that represents whether or not we want to skip odd numbers
- Within the method, print out numbers starting with 0 and counting up
  - Stop **AT** the value passed into the method
  - If skip odd numbers was chosen, only print the even numbers until you reach the designated number

METHOD REVIEW

- Write a method that takes in a number
  - If the number is even, return ‘E’
  - If the number is odd, return ‘O’
- Modify the previous method to now print ‘E’ and ‘O’ instead of the actual number