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

Method Header

Definition

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  
    return result;           user and adds them together
```

```
}
```

- 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

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);
```

```
}
```

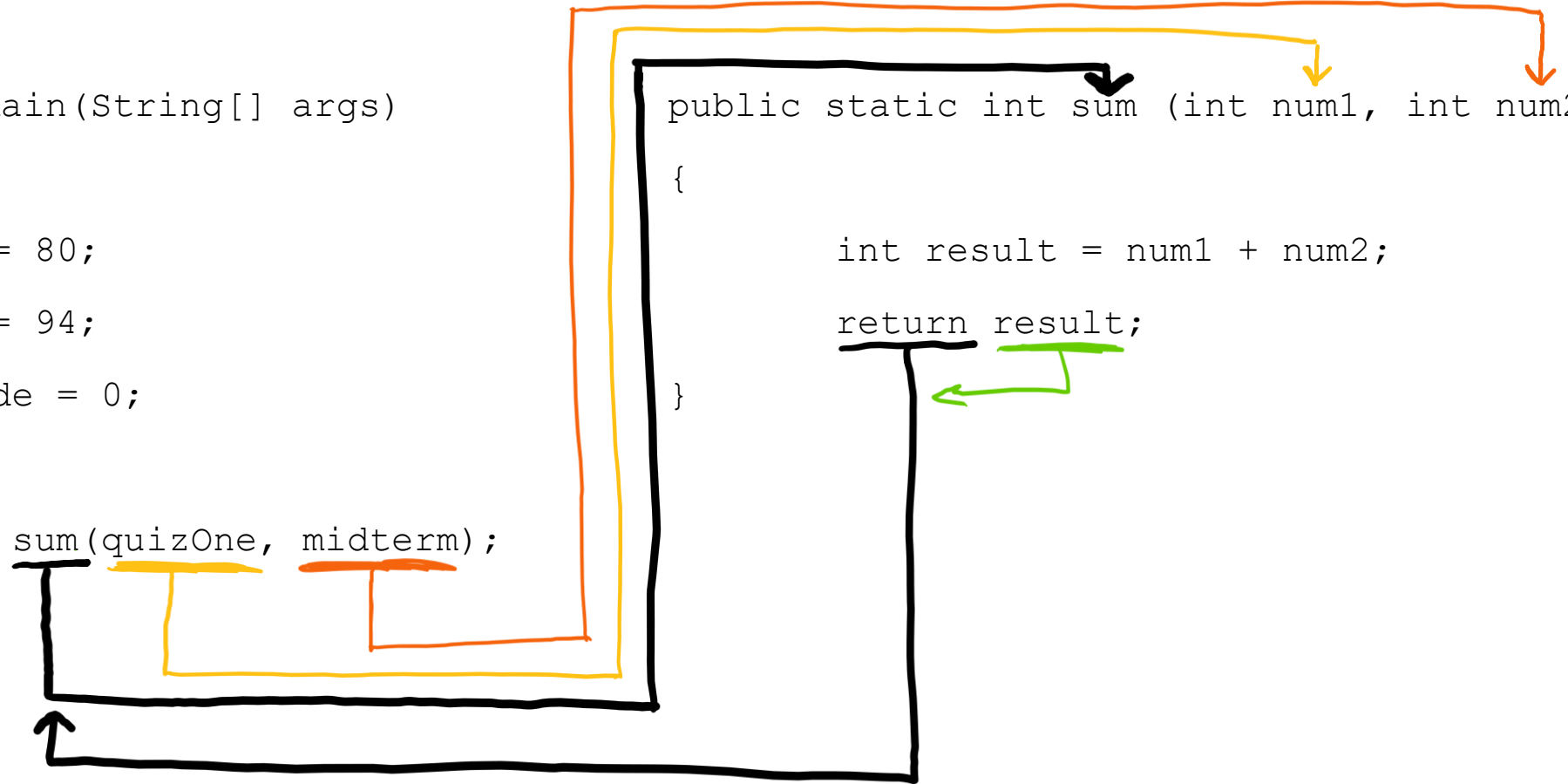
```
public static int sum (int num1, int num2)
```

```
{
```

```
    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
 - Call 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
 - Call by reference
 - Don't have to return something to make changes to the parameter