Methods

Upon completion of this module, a student will be able to

- write and call simple methods
- return a value from a method
- write methods that accept and use parameters
- learn to overload methods so that they be declared with many parameter combinations
- accept input from the user without a graphical user interface

Project

- Task
  - Write a guessing game where the user has to guess a secret number you will hard code as another data member. After every guess the program tells the user whether their number was too large or too small.
  - Must include these methods

    a. int checkGuess(int guess)

    i. check to see if the number is correct and return the result as an integer value

    b. void updateUI(int result)

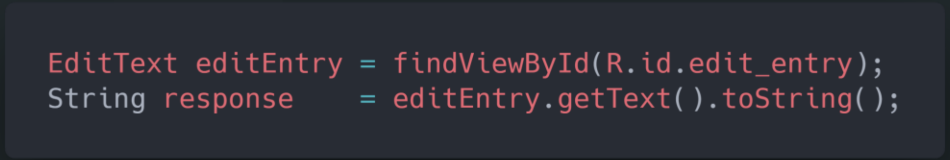
    i. this will perform all the tasks necessary to update the display once the result has been determined

    ii. use a switch statement inside of this method to determine what to display to the user based on the provided result.
- Repo
  - https://github.com/LambdaSchool/Android_Methods
- Submission
  - Zip up the project directory and send it to your PM
- Challenge
  - Randomly generate a secret number.
  - https://developer.android.com/reference/java/util/Random.html#nextInt(int)
  - When the user has correctly guessed the number, allow them to reset and allow the user to try again.
  - Experiment with attributes of your textViews and other GUI components to improve the look of your app

A Student Can

accept text input from the user

User Input



```
EditText editEntry = findViewById(R.id.edit_entry);
String response    = editEntry.getText().toString();
```

- EditText Component
- Editable Object
- Convert toString

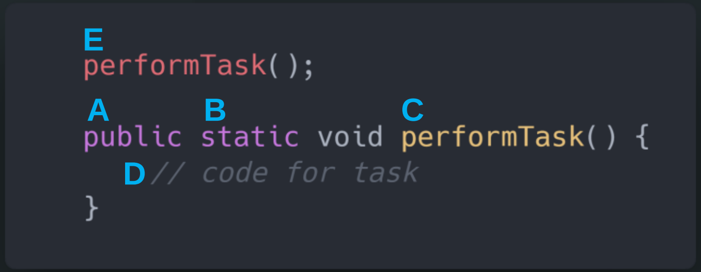
A Student Can

write and call simple methods

Methods

- sections of code that can be run from different places
- allows for modularization
- can return a result

Methods



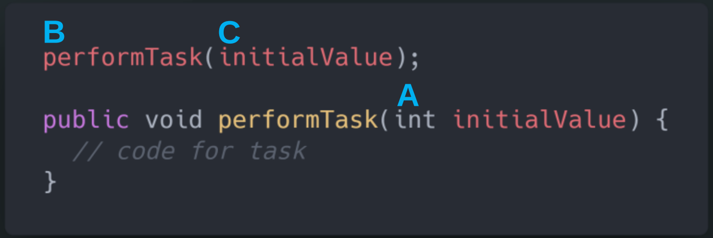
```
performTask();

public static void performTask() {
    // code for task
}
```

A. Visibility
B. static keyword
C. Name
D. Body
E. Method Call

A Student Can

write methods that accept and use parameters

Pass Parameters



```
performTask(initialValue);

public void performTask(int initialValue) {
  // code for task
}
```

A. Parameters

B. Method Call

C. Pass parameter

A Student Can

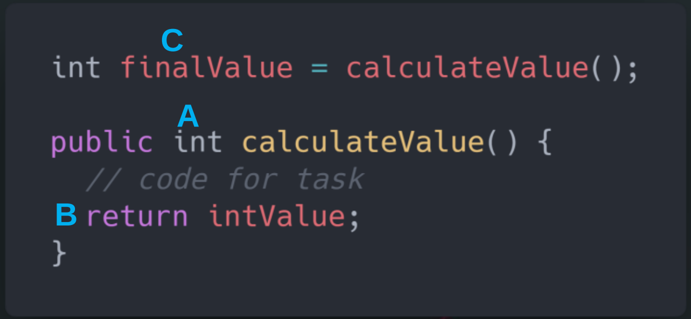
return a value from a method

Return Values

A. Return type

B. Return a value

C. Use the returned value



```
int finalValue = calculateValue();

public int calculateValue() {
  // code for task
 return intValue;
}
```

Challenge

- Write a method to find the smallest number among three numbers.

A Student Can

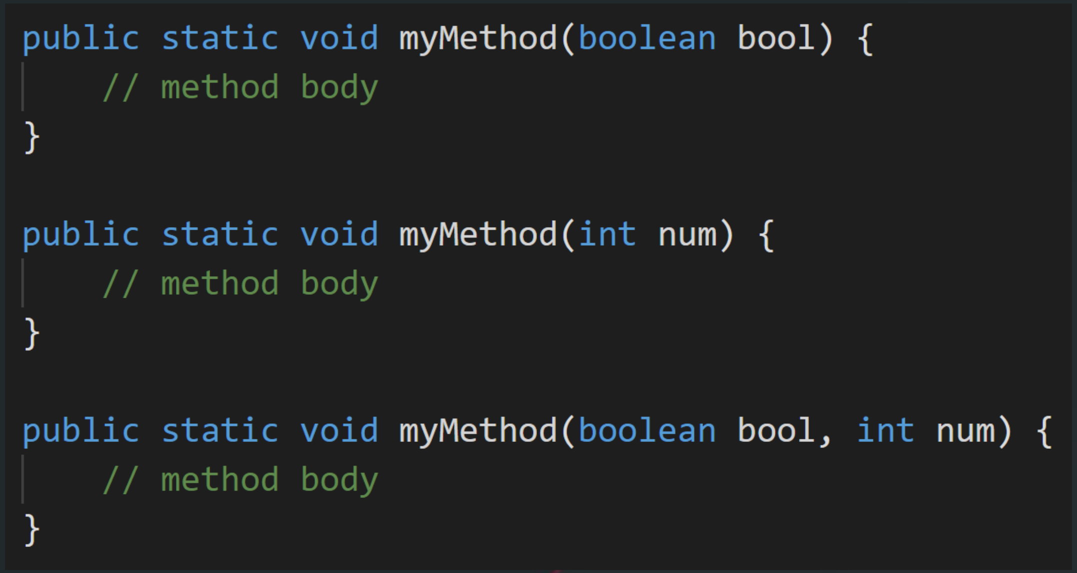
overload methods

Challenge

- Write a method to find the smallest number among three numbers, which can be passed in as Strings or ints.
- Hint: use Integer.parse(String) to convert from a String to an int;

Method Overloading

- Multiple methods with same name but different parameters

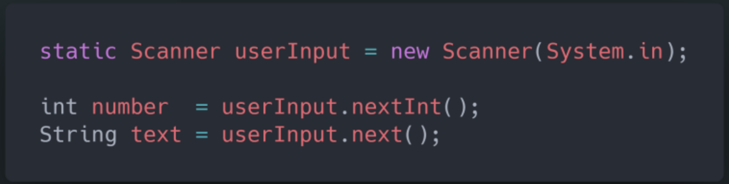


```java
public static void myMethod(boolean bool) {
    // method body
}

public static void myMethod(int num) {
    // method body
}

public static void myMethod(boolean bool, int num) {
    // method body
}
```

A Student Can

accept input from the console

User Input



```
static Scanner userInput = new Scanner(System.in);

int number  = userInput.nextInt();
String text = userInput.next();
```

- Scanner object
  - Give it a stream to scan
- use next() to get a string from the user
- use nextInt() to get a integer from the user