Computer Science 101 S1

Lecture 8

Contents

Practice with writing methods

void return type

Coursebook: §10

Ex01 - Define `getLongerLength()`

This method has two parameters both of type **`String`**. The method returns the length (an **`int`**) of whichever of the two parameter strings contains more characters.

```
public class CallingMethods {
  public void start() {
      int len1 = getLongerLengt 3  i", "BOO");
      String word = "stop now";
      int len2 = getLongerLengt 8  ery", word);
  }
  private       getLongerLength(               ){



  }
}
```

Ex02 - Define `getChange()`

The **`getChange()`** method has two parameters of type **`double`**. The first parameter is the amount paid and the second parameter is the cost. The method returns the difference between these two amounts. The value returned by the method is of type **`double`**.

```
public class CallingMethods {
  public void start() {
      double change1 = getCh 66.5 00, 33.5);
      double change2 = getC 27.5 50, 22.5);
  }
  private     getChange(                      ){



} }
```

Ex03 - Define `timeInMinutes()`

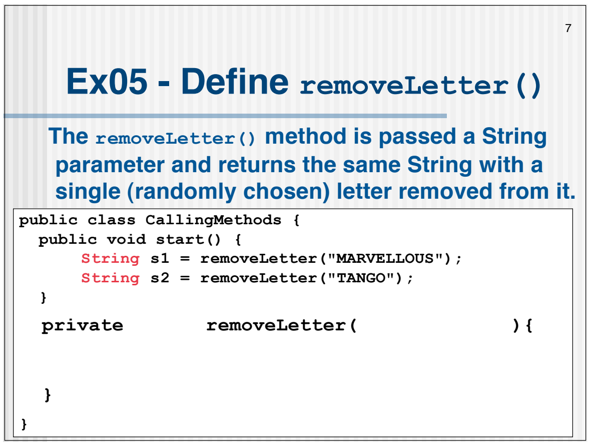
The **`timeInMinutes()`** method has two parameters both of type **`int`**. The first parameter is the number of hours and the second parameter is the number of minutes. The method returns the total number of minutes (an **`int`**) .

```
public class CallingMethods {
  public void start() {
      int mins1 = timeInMinutes(3, 40);
      int min2 = timeInMinutes(2, 15);
  }
  private     timeInMinutes(               ){



} }
```

Ex04 - Define `getRandom()`

The **`getRandom()`** method has one parameter of type **`int`**. The method returns a random number less than the parameter number (an **`int`**) .

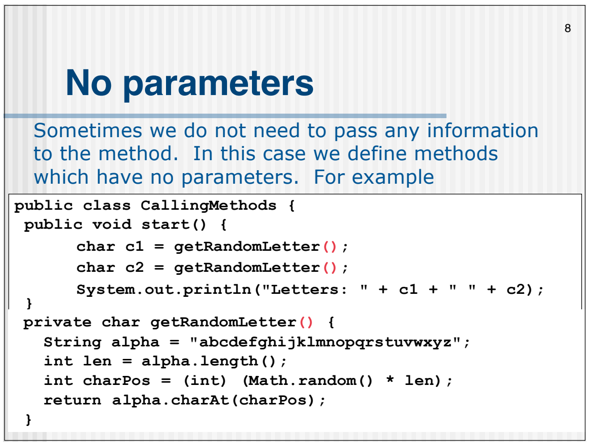
```
public class CallingMethods {
  public void start() {
      int num1 = getRandom(7);
      int num2 = getRandom(5);
  }
  private          getRandom(               ){


  }
}
```



Ex05 - Define `removeLetter()`

The `removeLetter()` method is passed a String parameter and returns the same String with a single (randomly chosen) letter removed from it.

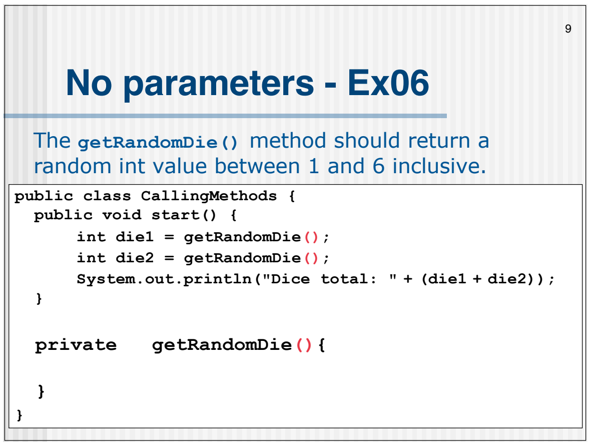
```
public class CallingMethods {
  public void start() {
      String s1 = removeLetter("MARVELLOUS");
      String s2 = removeLetter("TANGO");
  }
  private        removeLetter(               ){


  }
}
```



No parameters

Sometimes we do not need to pass any information to the method. In this case we define methods which have no parameters. For example

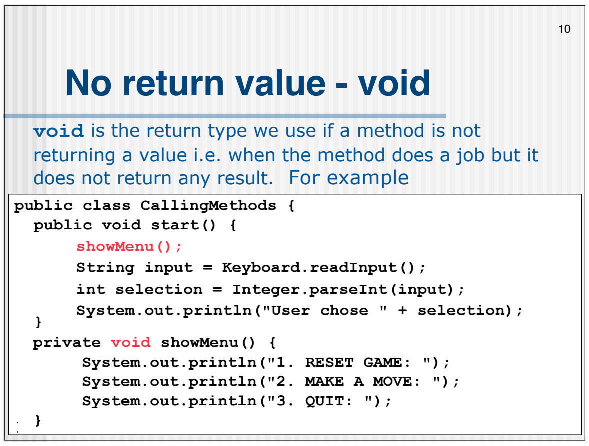
```
public class CallingMethods {
 public void start() {
      char c1 = getRandomLetter();
      char c2 = getRandomLetter();
      System.out.println("Letters: " + c1 + " " + c2);
 }
 private char getRandomLetter() {
   String alpha = "abcdefghijklmnopqrstuvwxyz";
   int len = alpha.length();
   int charPos = (int) (Math.random() * len);
   return alpha.charAt(charPos);
 }
```



No parameters - Ex06

The `getRandomDie()` method should return a random int value between 1 and 6 inclusive.

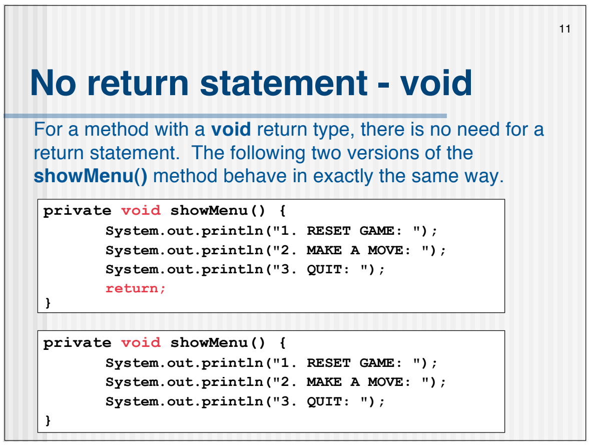
```
public class CallingMethods {
  public void start() {
      int die1 = getRandomDie();
      int die2 = getRandomDie();
      System.out.println("Dice total: " + (die1 + die2));
  }

  private   getRandomDie(){

  }
}
```



No return value - void

`void` is the return type we use if a method is not returning a value i.e. when the method does a job but it does not return any result. For example

```
public class CallingMethods {
  public void start() {
      showMenu();
      String input = Keyboard.readInput();
      int selection = Integer.parseInt(input);
      System.out.println("User chose " + selection);
  }
  private void showMenu() {
       System.out.println("1. RESET GAME: ");
       System.out.println("2. MAKE A MOVE: ");
       System.out.println("3. QUIT: ");
  }
```

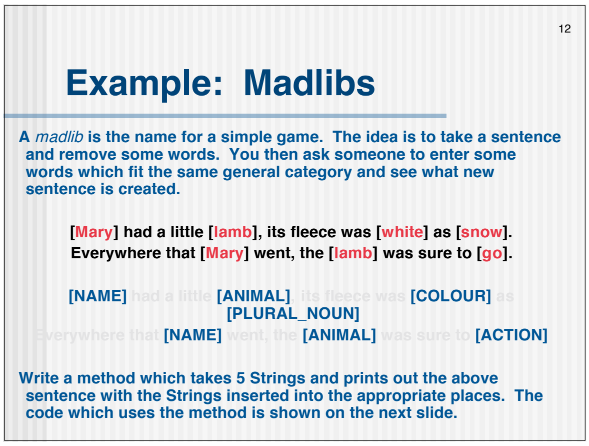


No return statement - void

For a method with a **void** return type, there is no need for a return statement. The following two versions of the **showMenu()** method behave in exactly the same way.

```
private void showMenu() {
        System.out.println("1. RESET GAME: ");
        System.out.println("2. MAKE A MOVE: ");
        System.out.println("3. QUIT: ");
        return;
}
```

```
private void showMenu() {
        System.out.println("1. RESET GAME: ");
        System.out.println("2. MAKE A MOVE: ");
        System.out.println("3. QUIT: ");
}
```



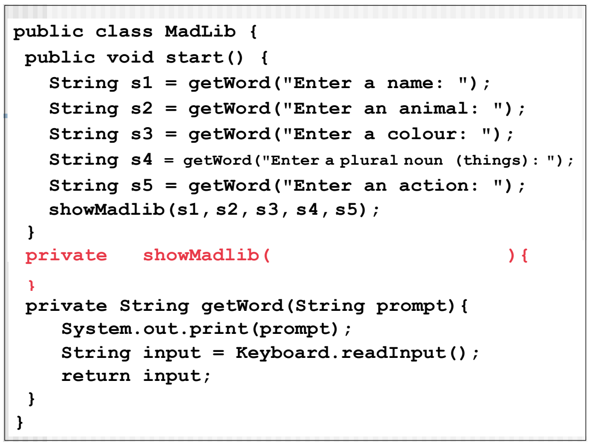
Example: Madlibs

A *madlib* is the name for a simple game. The idea is to take a sentence and remove some words. You then ask someone to enter some words which fit the same general category and see what new sentence is created.

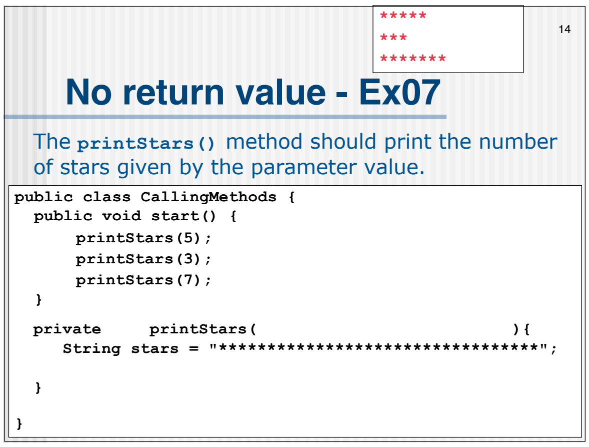
[Mary] had a little [lamb], its fleece was [white] as [snow]. Everywhere that [Mary] went, the [lamb] was sure to [go].

[NAME] had a little [ANIMAL], its fleece was [COLOUR] as [PLURAL_NOUN]
Everywhere that [NAME] went, the [ANIMAL] was sure to [ACTION]

Write a method which takes 5 Strings and prints out the above sentence with the Strings inserted into the appropriate places. The code which uses the method is shown on the next slide.



```
public class MadLib {
 public void start() {
   String s1 = getWord("Enter a name: ");
   String s2 = getWord("Enter an animal: ");
   String s3 = getWord("Enter a colour: ");
   String s4 = getWord("Enter a plural noun (things): ");
   String s5 = getWord("Enter an action: ");
   showMadlib(s1, s2, s3, s4, s5);
 }
 private    showMadlib(                    ){
 }
 private String getWord(String prompt){
    System.out.print(prompt);
    String input = Keyboard.readInput();
    return input;
 }
}
```

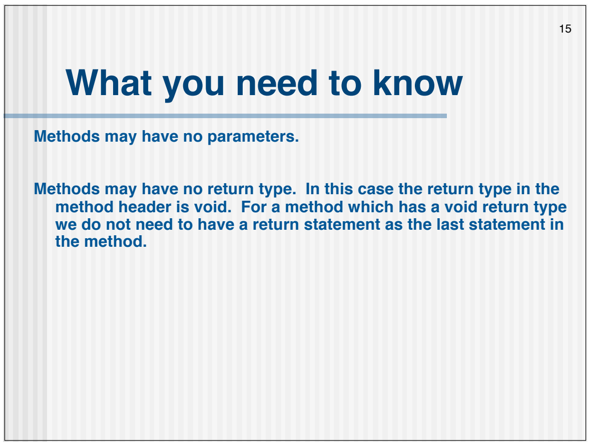


```
*****
***
*******
```

No return value - Ex07

The `printStars()` method should print the number of stars given by the parameter value.

```
public class CallingMethods {
  public void start() {
      printStars(5);
      printStars(3);
      printStars(7);
  }

  private     printStars(                           ){
     String stars = "***********************************";

  }

}
```



What you need to know

Methods may have no parameters.

Methods may have no return type. In this case the return type in the method header is void. For a method which has a void return type we do not need to have a return statement as the last statement in the method.