**Topic 1: Void Methods in Java**

Top of Form

Bottom of Form

**Main Method**

So far we have encountered one method in Java, it is the special main method. This method is called when the program starts. A Java program can consist of multiple classes, each in a separate file, but at least one class must contain a main.

public class HelloUser {

public static void main(String[] args) {

String userName = args[0];

System.out.println("Hello " + userName + "!");

}

}

The main method has a single parameter, the array String[] args. Last week we saw that the array will be populated with the command line arguments, for example, the above code prints the users name when it is passed on the command line.

**>** java HelloUser Jim

Hello Jim!

Your code will not compile and run without a main. It is case-sensitive, as many programming languages are. For example, a method called Main (capitalized) will give you the following error:

Error: Main method not found in class *classnamehere*, please define the main method as:  
   public static void main(String[] args)

**Void methods**

You may remember that when we first learned how to create pseudocode methods in Week 7, we started with methods that return nothing and have no parameters. A method that returns nothing is called a **void method**. In Java we explicitly use the word void before the name of the method when declaring it.

|  |  |
| --- | --- |
| **Pseudocodo** | **Java** |
| displayWelcome()    output "Welcome to guessing game."    output "Try your luck."  return | public static void displayWelcome() {    System.out.println("Welcome to the guessing game.");    System.out.println("Try your luck."); } |

You may be wondering about two other key words found in Java methods. The first one, **public**, we have seen before in Week 10, it indicates that code outside of the class can call this method.

The other key word, **static**, indicates that we do not need to create an object to call this method. This is handy because we have not learned how to create new objects in Java yet (we will in Topic 4). Usually programmers try to avoid making too many static methods because it is not good object-oriented programming (OOP) practice, but until we learn how to create objects we will declare all of our methods to be static.

We **call** a void method in the same way we did in psueodocode. One final note, methods always go inside a class. Here is an example of declaring and calling a method.

|  |  |
| --- | --- |
| **Pseudocodo** | **Java** |
| start  displayWelcome() stop  displayWelcome()    output "Welcome to guessing game."    output "Try your luck."  return | public class GuessGame {  public static void main(String[] args) {  dipalyWelcome();  }   public static void displayWelcome() {    System.out.println("Welcome to the guessing game.");      System.out.println("Try your luck.");   } } |

[**Void Methods**](https://dmacc.blackboard.com/webapps/assignment/uploadAssignment?content_id=_3812348_1&course_id=_69150_1&group_id=&mode=view)

Let's create a void method and call it from main. Here is what to do.

* 1. Write a void method that does not take any parameters called **printFavorite**. It should print your name and a favorite (subject, book, movie, etc) of yours.
  2. Call the method from main.

Run and test it. Submit your .java file.

This is worth 5 points.