**Finding Minimum or Maximum**

Next up is the Challenge Program. You will apply all of your Java knowledge acquired up to this point. Let's review some of the key concepts you have learned:

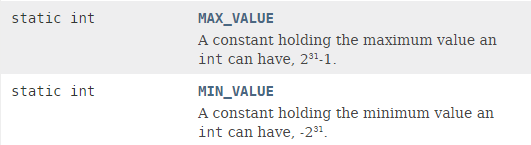
* printing **String** literals
* writing arithmetic statements
* using variables
* creating menus and requesting input
* changing the flow of control with decision statements
* using relational and logical operators
* managing the flow of control with loop statements
* generating random numbers
* reading and writing text files
* using arrays
* formatting output

You are gaining mastery over a powerful tool that will do your bidding as long as you follow its rules!

Before you begin the challenge program, you will need to be able to access a couple of Java's built-in constants. Constants are similar to variables, but their values cannot be changed. They can be useful in various situations, such as when you want to calculate the area of a circle. In that case, having a constant containing the value of Pi would be very helpful. For this project, knowing the minimum and maximum range for integers and doubles will be key.

### Part 1

The Integer class has two constants that are very useful: **Integer.MIN\_VALUE** and **Integer.MAX\_VALUE**. The Integer.MIN\_VALUE contains the smallest value that an integer variable can hold, and the Integer.MAX\_VALUE contains the largest value that an integer variable can hold.



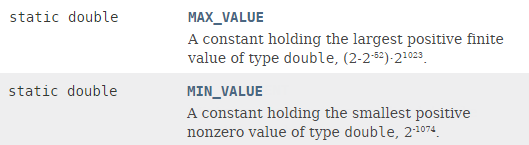
When finding the minimum value contained in an integer array, you can initially set the min variable to the Integer.MAX\_VALUE. As the array is traversed and values compared, the smallest value can be detected. Similarly, you can use set the max variable to Integer.MIN\_VALUE when you begin finding the maximum value within an integer array.

The following program finds the minimum and maximum values within a 5 integer array. Notice how the constants Integer.MAX\_VALUE and Integer.MIN\_VALUE are used.

 public class FindMinMax  
{  
    public static void main(String[] args)  
    {  
        int [] numList = {3, 99, -1, 5, -10};  
        int min = Integer.MAX\_VALUE;  
        int max = Integer.MIN\_VALUE;  
        for(int i = 0; i < numList.length; i++)  
        {  
            if (numList[i] < min)  
                min = numList[i];  
            if (numList[i] > max)  
                max = numList[i];  
        }  
        System.out.println("Minimum value is " + min);  
        System.out.println("Maximum value is " + max);  
    }     
}

Perform a quick Desk Check on this code. Make sure you understand each step.

1. Create a project called 06.05 Min and Max in the Mod06 Lessons folder.
2. Open the [FindMinMax.java](https://lti.flvsgl.com/flvs-cat-content/km6vpq52t1ffotgi1bt7mrrbov/flvs-cat-session/apcomputersciencea_v20/module07/lesson03/docs/07_03b/FindMinMax.java) file to the newly-created project.
3. What role did Integer.MAX\_VALUE have in this program. How about Integer.MIN\_VALUE?
4. Try printing the two constants. What do you notice about their values?
5. Convert the program to work with decimal values. The Double.MIN\_VALUE and Double.MAX\_VALUE constants are similar to the Integer class constants.



**This and That**

Have you noticed that instance variable names often begin with the ***my***-prefix? This is done to avoid confusion between instance variables and local variables, as the following two examples illustrate:

|  |  |  |
| --- | --- | --- |
| private double myName;  public Student(String name) {      myName = name; } |  | private double myName;  public void setName(String name) {       myName = name; } |

* The first example shows the Student constructor being passed a single parameter called name, which is assigned to an instance variable called myName.
* The second example shows the setName() method being passed a single parameter called name, which is assigned to an instance variable called myName.

Prefixing instance variable names is common practice, and you might even see prefixes like aName, anAmount, or theSum used. For novice programmers, using prefixed variable names is a useful habit to adopt, but it may eventually become annoying. Would you be surprised to learn that this convention is totally unnecessary?

### Part 1

Java permits instance variables and local variables to have the same name, but there may be unintended consequences if precautions are not taken to clarify which variable is being referred to. Several examples will help clear up the confusion:

* Open the [07.03 Virtual Lecture Notes](https://lti.flvsgl.com/flvs-cat-content/km6vpq52t1ffotgi1bt7mrrbov/flvs-cat-session/apcomputersciencea_v20/module07/lesson03/pop/07_03b/07_03b_popa.htm).
* Create a new project called 07.03 This Works in the Mod07 Lessons folder.
* Download the following Java files to the newly-created folder:   
  [ThisDemoATester.java](https://lti.flvsgl.com/flvs-cat-content/km6vpq52t1ffotgi1bt7mrrbov/flvs-cat-session/apcomputersciencea_v20/module07/lesson03/docs/07_03b/ThisDemoATester.java)   
  [ThisDemoBTester.java](https://lti.flvsgl.com/flvs-cat-content/km6vpq52t1ffotgi1bt7mrrbov/flvs-cat-session/apcomputersciencea_v20/module07/lesson03/docs/07_03b/ThisDemoBTester.java)   
  [ThisDemoCTester.java](https://lti.flvsgl.com/flvs-cat-content/km6vpq52t1ffotgi1bt7mrrbov/flvs-cat-session/apcomputersciencea_v20/module07/lesson03/docs/07_03b/ThisDemoCTester.java)
* Carefully read the discussion and carry out the instructions.

### Part 2

The keyword this is very versatile; it can be used with variables, objects, and constructors. Carefully study the interactive eIMACS exercises and labs in this section to understand how to use this effectively in programs.

[Complete the following eIMACS lab.](https://www.eimacs.com/)

* Object-oriented Programming
  + Class Definitions Revisited
    - Using this[1] [2] [3] [4]