

# MIT AITI Mobile Application Development in Java

## Lab 1: Skills Assessment Lab



Complete these problems and show your solutions to the instructors. We want to get a sense of your programming ability, so be prepared to explain how your code works.

We do not expect you to finish every problem, just do as many as you can. For each problem, we will provide you with an outline of the code you should write (“skeleton code”), called Number1.java, Number2.java... etc.

1. The code in Number1.java prints “Hello, World” to the screen. Modify it so that it prints “Hello, AITI”.
2. Add one line to Number2.java so that it prints  $9 \times y + 2$ . For example, the code should print 38 when y is 4.
3. The % operator returns the *remainder* of a number when divided by another number. For example,  $4 \% 3$  is 1 because 4 is 1 more than a multiple of 3. Similarly,  $9 \% 3$  is 0 because 9 is a multiple of 3. You can also use this operator with a variable, e.g.  $x \% 3$ .

Add if-then statements to the code in Number3.java so that it prints “Div3” if x is divisible by 3 ( $x \% 3$  is 0), “Div5” if x is divisible by 5, and “Div3Div5” if x is divisible by both 3 and 5. If it is divisible by neither 3 nor 5, just print the number. For example, if  $x = 17$ , you should print out 17, if  $x = 15$ , you should print out “Div3Div5.”

4. a) Use a for loop and the code from #3 to print out the following:

```
1
2
Div3
4
Div5
Div3
7
8
Div3
Div5
11
Div3
13
14
Div3Div5
```

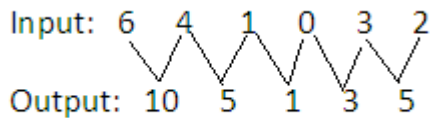
b) Use a nested loop to print out the following:

```
1
12
123
1234
12345
123456
```

For extra credit, solve this problem using only one loop (not a nested one).

- Write code that prints out the elements of an array separated by spaces. For example, {1,2,3,4} should be printed as 1 2 3 4
- a) Write code that “reduces” an array as in the following example:

Input: 6 4 1 0 3 2  
Output: 10 5 1 3 5



Where  $10 = 6 + 4$ ,  $5 = 4 + 1$ ,  $1 = 1 + 0$ ...

- Add to your code so that it reduces the array a second time.
  - Add to your code so that you reduce the array down to a single element.
- In Java, a `double` represents a floating point number (such as 3.141). Write a method that takes an array of `doubles` and a `double` as arguments. It should multiply every element in the array by that `double`. For example, if the input was {1.2,2.1} and 2.0, the output should be {2.4,4.2}. Don't worry if some of the numbers are slightly off—roundoff errors are common with `doubles`.
  - Add a method to the above code that takes an array of `doubles` and a `double` as arguments. It should multiply every *other* element in the array by that `double` (the first, third, fifth...). For example, if the input was {1.2,2.1,1.1} and 2.0, the output should be {2.4,2.1,2.2}.
  - Add a method to the above code that takes an array of `doubles` and two `doubles` as arguments. It should multiply the odd elements (first, third, fifth...) by the first `double` and the rest (second, fourth, sixth...) by the second `double`. For example, if the input was {1.2,2.1,1.1}, 2.0, and 3.0, the output should be {2.4,6.3,2.2}.

See if you can write the method in two lines (you may assume that neither of the `double` arguments is 0).

- Please fill out this survey once you're done with the lab: <http://bit.ly/12Blqqx>