

MIT AITI Mobile Application Development in Java

Lab 02: Java Basics



Instructions on how to submit Lab 2 will be presented on the screen. Please follow instructions and if they are unclear, please feel free to ask in class, or email us at: mustafanaseem@gmail.com & mahlet.woldeyes@gmail.com. Following client instructions is an important part of software development, and we will get more particular about this as the class progresses.

Section 2: Variables and Operators

Part I: Written Exercises (20 Minutes)

1. What is wrong with each of the following statements:

- a) `boolean isGood = 1;`
- b) `char firstLetter = p;`
- c) `int 2way = 89;`
- d) `String name = Ricky;`
- e) `int student score = 89;`
- f) `Double $class = 4.5;`
- g) `int _parents = 20.5;`
- h) `string name = "Greg";`

2. Without doing any programming, write down what the following main method prints to the screen?
(Hint: Keep track of the values of all the variables for each line)

```
public class UsingOperators {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        int z = x + x*y - y;
        System.out.println("The value of z is " + z);

        int w = (x + x)*y - y;
        System.out.println("The value of w is " + w);

        z = w + 3;
        System.out.println("The value of z is now " + z);

        z -= 2;
        System.out.println("The value of z is " + z);
    }
}
```

```

    z++;
    System.out.println("The value of z is " + z);

    --z;
    System.out.println("The value of z is " + z);

    boolean a = true;
    boolean b = false;
    boolean c = ((a && !(x > y)) && (a || y > x));
    System.out.println("c is " + c);
}
}
}

```

Part II: Computer Exercises

3. Create a new Java file called `UsingOperators` and copy the above code into it. Compile and run. Does the output match what you computed?
4. Create a new Java file called `TempConverter`. Add the following code to the file:

```

class TempConverter {
    public static void main(String[] args) {
        //place your code here
    }
}

```

Add code that declares and initializes a variable to store the temperature in Celsius. Your temperature variable should be stored numbers with decimal places. Which types would be appropriate to use for variables with decimal places?

5. Compute the temperature in Fahrenheit according to the following formula and print it to the screen. Note, $\text{Fahrenheit} = 9/5 * \text{Celsius} + 32$
6. Set the Celsius variable to 100 and compile and run `TempConverter`. The correct output is 212.0. If your output was 132, you probably used integer division somewhere by mistake.
7. Now we will create a program that will compute an average using two different variable types.

First create a Java file called `Fruit`. Type this code into the file.

```

class Fruit{
    public static void main(String[] args) {
        // Declare and initialize three variables here
    }
}

```

Initialize three variables:

1. An integer representing the number of fruits a boy is carrying.
2. An integer representing the number of fruits a girl is carrying.
3. The average number of fruits between them.

Declare the number of fruits the boy is carrying to be 5 and the number of fruits the girl is carrying to be 10.

Compute and print the precise average number of fruits between both people.

Section 3: Control Structures

Part I: Written Exercises

1. Consider the following code (draw a flowchart diagram if it helps):

```
if (x > 2) {  
    if (y > 2) {  
        int z = x + y;  
        System.out.println("z is " + z);  
    }  
}  
else {  
    System.out.println("x is " + x);  
}
```

What is the output if:

- a. $x = 2$ and $y = 5$?
 - b. $x = 3$ and $y = 1$?
 - c. $x = 1$ and $y = 1$?
 - d. $x = 4$ and $y = 3$?
2. What does the following code output, and how many times do we run through the *loop body*?

```
int i = 1;  
while (i < 10)  
{
```

```
        if ((i++) % 2 == 0)
        {
            System.out.println(i);
        }
    }
```

3. How about this version of the code?

```
int i = 1;
while (i > 10)
{
    if ((i++) % 2 == 0)
    {
        System.out.println(i);
    }
}
```

Part II: Computer Exercises

Create a new class called `UsingControlStructures` in your lab03 project. You will be submitting this java file, so be sure that everything works before you submit your lab. There will be instructions below to mark the different questions on this part of the lab.

At the very top of the file, insert

```
import java.util.*;
```

After the above line, copy the following code into your class:

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter an integer: ");
    int theInput = scanner.nextInt();

    //Your code here
}

```

You may not understand the details of the above code. Don't worry. It suffices to know that `scanner.nextInt()` waits for the user to type an integer and press the "Enter" key, then returns what they entered as an `int`. In the code we store the value in an integer variable called `theInput`.

4. Now insert code into the code above so that the program prints "even" if the input integer is even and "odd" if it is odd.
5. Suppose that we want to check if `theInput` is equal to 0, 1, or something else. If it is equal to 0, have the program print "Zero"; if it is equal to 1, have the program print "One"; otherwise, have the program print "Something else." Do not use an if statement
6. Write a program that sums the individual digits of an integer. For example, if an integer is 932, the sum is 14. Hint: Use the `%` operator to extract digits and use the `/` operator to remove the extracted digit. For instance, $932 \% 10 = 2$ and $932 / 10 = 93$.

Section 4: General (Variables+Operators; Control Structures; Arrays; Methods)

1. a) Write a method that takes an array of integers and another integer as arguments. If the integer is the sum of two different elements of the array, return `true`. Otherwise, return `false`.

For example, if your inputs were `{1, 3, 6}` and 9, the method would return `true` since $9 = 6 + 3$. If your inputs were `{1, 3, 6}` and 10, the method would return `false`. If your inputs were `{1, 3, 6}` and 6, your method would return `false`: even though $6 = 3 + 3$,

you want to use *distinct* elements of the array.

You may assume that the array has no duplicated values (no arrays like $\{1, 3, 6, 6\}$).

- b) Extend the previous problem so that your method has two `int` inputs. The second `int` input determines the number of (*distinct*) elements that should be added to make the first integer input.

For example, if your inputs were $\{1, 3, 6, 7\}$, 11, and 3, your method would return `true` as $11 = 1 + 6 + 7$. If your inputs were $\{1, 3, 6, 7\}$, 17, and 4, your method would return `true` as $17 = 1 + 3 + 6 + 7$.

- Write a recursive program that extends the range of the Fibonacci sequence. The Fibonacci sequence is 1 1 2 3 5 8 etc. where each element is the sum of the previous two elements. For this problem, instead of only adding the last two elements to get the n th element, have the n th element be the sum of the previous k elements. For example, the 7th Fibonacci element summing the previous 3 elements would be 37. The sequence is 1 1 2 4 7 13. Be sure to check your code by running it with a k value of 2 and comparing the value to the value you would obtain with the regular Fibonacci sequence (they should be the same).
- Write a program that computes the factorial of a non-negative integer. For example, the factorial of 4 or $4!$ is $4 \times 3 \times 2 \times 1 = 24$. If you understand recursion, you may use that to write your program.
- Write a program that will test to see whether or not a given `String` is a palindrome. A palindrome is a word or phrase which reads the same in both directions. examples: racecar; level; dad; mom; madam; If a word is a palindrome, print out a statement saying it is. If not, then print out a statement saying it is not.
- Create a Java file named `Lab02_prime.java` that displays the first fifty *prime* numbers in five lines, each line contains 10 numbers. An integer greater than 1 is prime if its only positive divisor is 1 or itself. For example, 2, 3, 5, and 7 are prime but 4, 6, 8, and 9 are not prime. The output of your program should look like:

```
The first 50 prime numbers are 2 3 5 7 11 13 17 19 23 29 31 37 41
43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131
137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223
227 229
```

- Use nested loops to print out each of the following patterns. Create a separate Java file for each pattern named `Lab02_6a.java`, `Lab02_6b.java` and `Lab02_6c.java`

```

a. 1 2 3 4 5 6
    1 2 3 4 5
    1 2 3 4
    1 2 3
    1 2
    1

```

```

b.           1
          2 1
        3 2 1
       4 3 2 1
      5 4 3 2 1
     6 5 4 3 2 1

```

Extra Credit:

c. Write nested loops that will print the following pattern:

```

                                1
                              1  2  1
                            1  2  4  2  1
                          1  2  4  8  4  2  1
                        1  2  4  8 16  8  4  2  1
                      1  2  4  8 16 32 16  8  4  2  1
                    1  2  4  8 16 32 64 32 16  8  4  2  1
                  1  2  4  8 16 32 64 128 64 32 16  8  4  2  1

```

Reproduce the pattern exactly; note the spacing and how the digits align between different lines.

7. Add a method merge that takes two arrays as input and outputs a new array that contains their combined elements:

```
public static double[] merge(double[] a, double[] b)
```