Lab 2 Java Programming
Part 1: In-class Exercises

Writing Good Programs

- Coding style:
  - Follow the Java Naming Conventions for variables, methods, and class STRICTLY.
  - See lecture notes if you don't remember.
- Program documentation:
  - Comment! Comment! and more Comment!

The only way to learn programming is program, program and program.

Outline

- Data Types (Beginner Level)
- Flow Controls (Beginner Level)
- String and Char Operations (Beginner Level & Intermediate Level)
- Arrays (Advanced Level)

Organizing Projects

- Use the **`File → Switch Workspace`** menu in Eclipse to create a new workspace named "**`Lab2Part1`**" on the desktop of your computer.
- For each question in Part 1 of Lab 2, create a new project in that workspace. Call each project by its question number: "**`Question1`**", "**`Question2`**", etc.

Data Types

Question 1: write a Java program called **`InchesToMeters`** which reads a number in inches and converts it to meters.

Note: 1 inch = 0.0254 meter.

Test Data:

```
Input a value for inches: 1000.0
```

Expected Output:

```
1000.0 inches is 25.4 meters
```

Hints:

```
import java.util.Scanner;

public class InchesToMeters {
    public static void main(String[] args) {
        System.out.print("Input a value for inches: "); // hint for input
        Scanner input = new Scanner(System.in); // create a Scanner
        double inch = input.nextDouble(); // obtain the input value
        ....
    }
}
```

Data Types

Question 2: write a Java program called **`SumOfDigits`** which reads an integer between 0 and 1000 and adds all the digits in the integer.

Test Data:

```
Input an integer between 0 and 1000: 565
```

Expected Output:

```
The sum of all digits in 565 is 16
```

Flow Controls

- Conditional (Decision)
- Loop (Iteration)

Flow Controls -- Decision

Question 3: write a program called **`CheckPassFail`** which prints "**`PASS`**" if the integer variable "**`mark`**" is more than or equal to **`50`**; or prints "**`FAIL`**" otherwise. The program must always print "**`DONE`**" before exiting.

Flow Controls -- Decision

Question 4: write a program called **`CheckOddEven`** which prints "**`Odd Number`**" if the integer variable "**`number`**" is odd, or "**`Even Number`**" otherwise. The program must always print "**`BYE!`**" before exiting.

Flow Controls -- Decision

Question 5: write a program called **`PrintNumberInWord`** which prints "**`ONE`**", "**`TWO`**",... , "**`NINE`**", or "**`OTHER`**" if the integer variable "**`number`**" is **`1`**, **`2`**,... , **`9`**, or some other value, respectively.

The result must be printed twice:

- once using a "nested-if" statement;
- and once using a "switch-case" statement.

Flow Controls -- Loop

Question 6: write a program called **`SumAndAverage`** to produce the sum of **`1`**, **`2`**, **`3`**, ..., up to **`100`**. Use a "**`for`**" loop. Also compute and display the average. The output must look like:

```
The sum is 5050
The average is 50.5
```

6.1: modify the program to use a "**`while`**" loop instead of "**`for`**" loop.

6.2: modify the program to use a "**`do-while`**" loop.

6.3: modify the program to sum from **`111`** to **`8899`**, and compute the average.

String and char Operations

Question 7 (beginner level): write a program called **`ReverseString`** which prompts the user for a string, and prints the *reverse* of the string. The output must look like:

```
Enter a string: abcdef
The reverse of the string "abcdef" is "fedcba"
```

Hint: to read a string, create a **`Scanner`** object called **`input`** (see Question 1) and then use **`input.next()`**. For a string called **`inStr`**, you can use **`inStr.length()`** to get the *length* of the string, and **`inStr.charAt(index)`** to retrieve the character at the **`index`** position, where **`index`** starts at **`0`**.

String and char Operations

Question 8 (intermediate level): a word that reads the same forward and backward is called a *palindrome*, e.g., "mom", "dad", "racecar", "madam", and "Radar" (case-insensitive). Write a program called **TestPalindrome**, that asks the user for a word and prints whether the word is a palindrome or not.

Hint: for a string called **`inStr`**, you can use **`inStr.toLowerCase()`** which returns a new string which is all in lower case letters. Use two indexes **`forwardIndex`** and **`backwardIndex`** to scan the string forward and backward at the same time.

Array

Question 9: write a program called **`GradesAverage`**, which asks the user for a number of students and saves it in an integer variable called **`numStudents`**. It then asks the user for the grades of each of the students and saves them in an integer array called **`grades`**. Your program must check that the grades are between **`0`** and **`100`** and it must compute the average of the grades at the end. A sample session is as follow:

```
Enter the number of students: 3
Enter the grade for student 1: 55
Enter the grade for student 2: 105
Invalid grade, try again...
Enter the grade for student 2: 56
Enter the grade for student 3: 57
The average is: 56.0
```

Hint: after reading the value of **`numStudents`** from the user, use "**`new int[numStudents]`**"to create an array of integers of the right size.

Lab 2 Java Programming
Part 2: Homework Exercises

Organizing Projects

- Use the **`File → Switch Workspace`** menu in Eclipse to create a new workspace named "**`Lab2Part2_1234567890`**" on the desktop of your computer (replace **`1234567890`** with your student ID number).
- For each question in Part 2 of Lab 2, create a new project in that workspace. Call each project by its question number: "**`Question1`**", "**`Question2`**", and "**`Question3`**".

Lab 2 Homework Exercises

Question 1: copy-paste the **`SumAndAverage`** program from the sample answer of Question 6 of Part 1. Then:

- Modify the "**`for`**" loop to sum only the *odd* numbers from **`1`** to **`100`**, and compute the average.
- Modify the "**`while`**" loop to sum only the numbers from **`1`** to **`100`** that are divisible by **`7`**, and compute the average.
- Modify the first "**`do-while`**" loop to find the "sum of the squares" of all the numbers from **`1`** to **`100`**, i.e. 1*1 + 2*2 + 3*3 + ... + 100*100, and print the result.
- Delete all the code for the old Question 6.3 (the second "**`do-while`**" loop and anything it prints).

Lab 2 Homework Exercises

Question 2: write a program called **`TimeTable`** that prints the multiplication table of **`1`** to **`9`** as shown below, using two nested "**`for`**" loops:

```
*  |   1   2   3   4   5   6   7   8   9
---------------------------------------
1  |   1   2   3   4   5   6   7   8   9
2  |   2   4   6   8  10  12  14  16  18
3  |   3   6   9  12  15  18  21  24  27
4  |   4   8  12  16  20  24  28  32  36
5  |   5  10  15  20  25  30  35  40  45
6  |   6  12  18  24  30  36  42  48  54
7  |   7  14  21  28  35  42  49  56  63
8  |   8  16  24  32  40  48  56  64  72
9  |   9  18  27  36  45  54  63  72  81
```

The first two lines can be printed directly, there is no need to compute them using a loop.

Lab 2 Homework Exercises

Question 3: copy-paste the **`TestPalindrome`** program from the sample answer of Question 8 of Part 1. Then modify the program to check a whole *sentence* (not just one word) to see whether it is a palindrome or not. Punctuation, spaces, and capitalization must be ignored.

Here are examples of palindromic sentences:

- Madam, I'm Adam
- A man, a plan, a canal - Panama!

Hint: to read a whole *sentence*, create a **`Scanner`** object called **`input`** (see Question 1 of Part 1) and then use **`input.nextLine()`**. For a character variable called **`c`**, you can use **`Character.isLetter(c)`** to compute a boolean result indicating whether the character is a letter or not.

Lab 2 Submission

- Make sure you put comments everywhere in your code or you will lose points.
- After you have finished the assignment (creating three Eclipse projects, one for each question), exit Eclipse.
- Create a ZIP archive of the whole workspace folder. The resulting ZIP file must be called "**`Lab2Part2_1234567890.zip`**" (replace **`1234567890`** with your student ID number).
- Upload the ZIP file on iSpace.