**CS324: Advanced Programming in Java**

**Homework No. 4**

**Due on Friday 1st October.**

**Question No. 1**

**Number Parser:**

The following code contains a program that does the following: ­

* Prompts for and reads in a line of input ­
* Uses a second Scanner to take the input line one token at a time and parses an integer from each token as it is extracted. ­
* Sums the integers. ­
* Prints the sum.

If you compile and run it, and give it the input

10 20 30 40

it should print The sum of the integers on the line is 100. Try some other inputs as well. Now try a line that contains both integers and other values, e.g.,

We have 2 dogs and 1 cat.

You should get a NumberFormatException when it tries to read "We", which is not an integer. One way around this is to put the loop that reads inside a try and catch the NumberFormatException but not do anything with it. This way if it's not an integer it doesn't cause an error; it goes to the exception handler, which does nothing.

Do this as follows: ­ Modify the program to add a try statement that encompasses the entire while loop. The try and opening { should go before the while, and the catch after the loop body. Catch a NumberFormatException and have an empty body for the catch. ­

Compile and run the program and enter a line with mixed integers and other values. You should find that it stops summing at the first non-integer, so the line above will produce a sum of 0, and the line "1 fish 2 fish" will produce a sum of 1. This is because the entire loop is inside the try, so when an exception is thrown the loop is terminated.

To make it continue, move the try and catch inside the loop. Now when an exception is thrown, the next statement is the next iteration of the loop, so the entire line is processed. The dogs-and-cats input should now give a sum of 3, as should the fish input.

import java.util.Scanner;

public class ParseInts

{

    public static void main(String[] args)

    {

        int val, sum=0;

        Scanner scan = new Scanner(System.in);

        String line;

        System.out.println("Enter a line of text");

        Scanner scanLine = new Scanner(scan.nextLine());

        while (scanLine.hasNext())

        {

            val = Integer.parseInt(scanLine.next());

            sum += val;

        }

        System.out.println("The sum of the integers on this line is " + sum);

    }

}

**Question No. 2**

**Passport**

Create a class Passport, which have following attributes: Name (string), passport number (string), issue date (Date), expiry date (Date).

Define a constructor in this class that takes the arguments (String name, String passportnumber, String issuedate, String expiryDate). The constructor can throw the exceptions InvalidPassportNumberException, and DatesNotValidException.

InvalidPassportNumberException: Pakistani passport format is AB123456, i.e. first two letters are alphabets, and remaining six letters are number. In the constructor you should check if the format is correct, else you raise this exception.

DatesNotVaidException: Logically speaking, the issue date should be lesser than expiry date of a document. So, the constructor should check this condition and raise this exception if the dates are not correct. You can convert a string to a Date object in java, refer to this tutorial: <https://www.tutorialspoint.com/java/java_date_time.htm>