LAB 01

Exception and Assertion

java.lang

A. REQUIREMENTS

I. Create an Exception named CustomException return a custom error message.

II. Write TestThrowException class have number field (integer type).

- Add a method named setNumber (int pNumber) to set number value. If number <= 0, throw an CustomException exception
- Add main method to test program

III. Write a assertion program to check number > 0

IV. Write a program to display following information.

- All of Environment properties
- Name of current Operation System
- Java version
- Maximum memory can be used.
- Available processors
- Executes Notepad, MS Paint programs

V. ***Class*** Class

- Design a class named Booking for booking the airline tickets.
- Modify the class to store the following details:
  - departureDate
  - numberOfTickets
  - price
  - cabinType
  - totalPrice
  - destination
- Add appropriate constructors in the class to instantiate an object of the Booking class. The constructor initializes the field bookingID, departure/date, numberOfTickets, price, destination, and cabinType respectively.
- Create a method ticketConformation() to display all the information about the tickets booked.
- Use of some of the methods of Class class to reflect information about: fields, methods with public modifier.

VI. Write a program to demonstrate Regular Expressions in Java. Initialize a String like String candidate = "AAA99SuperJava" and display the output as a group like this

```
AAA
99
SuperJava
```

VII. Write a regex program to check valid date format from user input. Valid format is dd/MM/yyyy (ex: 24/02/2012 or 24/2/2012, 04/12/2012 or 4/12/2012)

B. HINTS

I. Custom exception

```
public class CustomException extends Exception {
    String message;

    public CustomException(String message) {
        this.message = message;
    }

    @Override
    public String getMessage(){
        return message;
    }

}
```

II. Test throw exception

```
public class TestThrowException {
    int number;
    void setNumber(int pNumber) throws CustomException{
        if(pNumber <= 0){
            throw new CustomException("Number must be greater then
0");
        }
        number = pNumber;
    }

    public static void main(String[] args) {
        TestThrowException obj = new TestThrowException();
        try {
            obj.setNumber(-2);
        } catch (CustomException ex) {

Logger.getLogger(TestThrowException.class.getName()).log(Level.SEVER
E, null, ex);
        }
    }
}
```

III. Test assertion

```
public class TestAssertion {
    public static void main(String[] args) {
        int a;
        Scanner input = new Scanner(System.in);
        System.out.println("Please enter number a: ");
        a = input.nextInt();

        assert (a > 0 && a < 10):"The number must be greater than 0
and less than 10";
        System.out.println("Number is ok!");
    }
}
```

IV. Use java.lang

```
System.out.println("*** All properties: ---");
Properties p = System.getProperties();
p.list(System.out);

System.out.println("*** Current OS: " + System.getProperty("os.name"));
System.out.println("*** Current Java Version: " + System.getProperty("java.version"));
System.out.println("*** Maximum memories: " + Runtime.getRuntime().maxMemory());
System.out.println("*** Available processors: " + Runtime.getRuntime().availableProcessors());
System.out.println("*** Executes programs: ");
try {
    Runtime.getRuntime().exec("NotePad.exe");
    Runtime.getRuntime().exec("mspaint");
} catch (IOException ex) {
    Logger.getLogger(DisplayInformation.class.getName()).log(Level.SEVERE, null, ex);
}
```

VI.

B1: Tạo mẫu Pattern:

```
Pattern pt = Pattern.compile("...");
```

B2: Tạo đối tượng Matcher để so khớp

```
Matcher mc = pt.matcher("AAA99SuperJava");
```

B3: Tìm từng nhóm phần tử hợp lệ và in ra.

```
while(mc.find()){
      System.out.println(mc.group());
}
```

V.

Gợi ý

```
(0?[1-9]|[12][0-9]|3[01])/(0?[1-9]|1[012])/((19|20)\\d\\d)
```

Giải thích:

(	Bắt đầu nhóm thứ nhất
0?[1-9]	01-09 hoặc 1-9
|	Hoặc
[12][0-9]	10-19 hoặc 20-29
|	Hoặc
3[01]	30 hoặc 31
)	Kết thúc nhóm thứ nhất
/	Dấu /
(	Bắt đầu nhóm thứ hai
0?[1-9]	01-09 hoặc 1-9
|	Hoặc
1[012]	10 hoặc 11 hoặc 12
)	Kết thúc nhóm thứ hai
/	Dấu /
(	Bắt đầu nhóm thứ ba
(19|20)\\d\\d	19[0-9][0-9] hoặc 20[0-9][0-9]
)	Kết thúc nhóm thứ ba