**Internet programming (Lab 2)**

**Problem 1: [10]**

Write a Java program takes as input from user an 8-digit string for a birth date. The first two digits in the string are the month of birth, the next two are the day and the remaining four are the year. The Java program should split the string into substrings and display it on the screen. You need to check if the month and day are out of range, and handle the exceptions of Integer.parseInt() e.g. if user enters alphabets instead of numbers.

* **Suppose the string entered is: 07211979**  
  Then prints:  
    
  month of birth: 07  
  day of birth: 21  
  year of birth: 1979
* **Suppose the string entered is: 06311979**  
  Then prints:  
    
  Month 06 only has only 30 days. Try again.

**Hints**:

* Use the String class and the method public String substring(int beginIndex,int endIndex).A sample of using substring method is given below.
* Similarly Integer.ParseInt() takes a string as input and converts that string to integer (if it is valid).

|  |
| --- |
| **class** Test  {  **public** **static** **void** main(String ar[])  {    String Str = **new** String("mystring");  System.***out***.print("Return Value :" );  System.***out***.println(Str.substring(3, 5) );  }  } |

**Problem 2: [15]**

Java provides quite a few exception classes. Use them whenever possible instead of creating your own exception classes. However, if you run into a problem that cannot be described by the predefined exception class, you can create your own exception class, derived from Exception or a subclass of IOException.

In this task you have to create your own exception class and handle that as well.

You are given account class.

|  |
| --- |
| **class** Account{  **private** **int** id;  **private** **double** balance;  **public** Account(**int** id, **double** balance){  **this**.id = id;  **this**.balance = balance;  }  **public** **int** getId(){  **return** id;  }  **public** **void** setBalance(**double** balance){  **this**.balance = balance;  }  **public** **double** getBalance(){  **return** balance;  }  //deposit an amount to this account  **public** **void** deposit(**double** amount)  {  **if** (amount >= 0)  balance = balance + amount;  }  //withdraw an amount from this account  **public** **void** withdraw(**double** amount){  **if** (amount >= 0)  balance = balance - amount;  }  } |

1. You have to create your own exception class named “NegavtiveBalanceException” (inherit it from Exception class), and **throw** exception of type “NegavtiveBalanceException” both in deposit and withdraw method if amount is less than zero. (**DO NOT catch it in those methods**)

Also write main method to demonstrate account class methods and handle (catch) exception when amount <0 is passed to withdraw and deposit methods. [10]

1. Change parent of your exception class from “Exception” to “IOException”. Does this make any difference? If yes what is that and what changes you’ll make to your program? [5]

**Problem 3: [10]**

1. Open Mysql (through commandline)
2. Create a database named MyDB. (***command:*** *create databse BankDB*)
3. Create a table named “**Accounts**” in MyDB. (
4. Accounts table should have 2 columns

**Coumnn1 Name:**ID, **Column1 type:** int

**Coumnn2 Name:** username, **Column2 type:** varchar(50)

**Column3 Name:** password, **Column3 type:** varchar(50)

(***command:*** *create table Accounts (id int, username varchar(50), password varchar(50))*)

Run the following query to add some data to accounts table:

|  |
| --- |
| insert into Accounts (id, username, password) values (1, ‘user1’, ‘password1’);  insert into Accounts (id, username, password) values (2, ‘user2’, ‘password2’);  insert into Accounts (id, username, password) values (3, ‘user3’, ‘password3’); |

Now you create a java application:

In main method, you have to establish connection with database then

* get all the users from accounts table and display it on console.
* Take a name as input from user and display the password for that username ( hint: use where clause in select).