Write a simple program which loops over a list of user data (tuples containing a username, email and age) and adds each user to a directory if the user is at least 16 years old. You do not need to store the age. Write a simple exception hierarchy which defines a different exception for each of these error conditions:

* the username is not unique
* the age is not a positive integer
* the user is under 16
* the email address is not valid (a simple check for a username, the @ symbol and a domain name is sufficient)

Raise these exceptions in your program where appropriate. Whenever an exception occurs, your program should move onto the next set of data in the list. Print a different error message for each different kind of exception.

You can consider an email address to be valid if it contains one @ symbol and has a non-empty username and domain name – you don’t need to check for valid characters. You can assume that the age is already an integer value.

User.java

public class User {

private int id;

private String userName;

private String emailId;

private int age;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getEmailId() {

return emailId;

}

public void setEmailId(String emailId) {

this.emailId = emailId;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public void checkUserName(User user[],int index){

for (int i=0;i<index;i++){

if(user[index].getUserName().equals(user[i].getUserName())){

throw new CustomException("Username already Exists");

}

}

}

public void checkAge(int age){

if(age<0){

throw new CustomException("Enter positive value for age");

}else if(age<16){

throw new CustomException("Age under 16");

}

}

public void checkEmailId(String email){

if(!email.contains("@")){

throw new CustomException("Invalid email");

}

if(!email.contains(".")){

throw new CustomException("Invalid email");

}

}

}

TestUser.java

import java.util.Scanner;

public class TestUser{

public static void main(String[] args) {

String userName,emailId;

int age,count;

Scanner sc=new Scanner(System.*in*);

User[] user=new User[50];

System.*out*.println("\nEnter the number of user:\t");

count=sc.nextInt();

for(int i=0;i<count;i++){

user[i]=new User();

System.*out*.println("\nEnter the "+(i+1)+" user name:\t");

userName=sc.next();

user[i].setUserName(userName);

System.*out*.println("\nEnter the "+(i+1)+" user email id:\t");

emailId=sc.next();

user[i].setEmailId(emailId);

System.*out*.println("\nEnter the "+(i+1)+" user age:\t");

age=sc.nextInt();

user[i].setAge(age);

try{

user[i].checkAge(user[i].getAge());

user[i].checkUserName(user, i);

user[i].checkEmailId(user[i].getEmailId());

}catch(Exception e){

System.*out*.println(e);

}

}

}

}

CustomException.java

public class CustomException extends RuntimeException{

public CustomException(String message){

super(message);

}

}