Name: Shangirne Kharbanda

Registration Number: 20BAI1154

JAVA LAB-5

EXCEPTION HANDLING EXERCISES

1. Write an application named BadSubscriptCaught in which you declare an array of eight first names. Write a try block in which you prompt the user for an integer and display the name in the requested position. Create a catch block that catches the potential ArrayIndexOutOfBoundsException thrown when the user enters a number that is out of range. The catch block also should display an error message. Save the file as BadSubscriptCaught.java.

BadSubscriptCaught.java

```
import java.util.Scanner;
public class BadSubscriptCaught {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n1;
        System.out.println("Enter the number of Strings you wanna enter:");
        n1 = sc.nextInt();
        sc.nextLine();
        String[] names = new String[n1];
        System.out.println("Enter the " + n1 + " number of strings:");
        for(int i = 0;i<n1;i++){
            names[i] = sc.nextLine();
        }
        try {
            System.out.println("Enter number: ");
            int n = sc.nextInt();
            n=n-1;
            System.out.println(names[n]);
        }
        catch (ArrayIndexOutOfBoundsException q)
        {
            System.out.println("Please enter number between 1 to " + n1);
        }
    }
}
```

```
Enter the number of Strings you wanna enter:
6
Enter the 6 number of strings:
Harry
Max
Rupert
Whatever
Ronnie
Radke
Enter number:
8
Please enter number between 1 to 6

Process finished with exit code 0
```

```
Enter the number of Strings you wanna enter:
5
Enter the 5 number of strings:
Harry
Potter
Maggie
Something
Nothing
Enter number:
3
Maggie

Process finished with exit code 0
```

2. Write an application that throws and catches an ArithmeticException when you attempt to take the square root of a negative value. Prompt the user for an input value and try the Math.sqrt() method on it. The application either displays the square root or catches the thrown Exception and displays an appropriate message. Save the file as SqrtException.java.

SqrtException.java

```
import java.util.*;
import java.lang.Math;

public class SqrtException {
    public static void main(String[] args) {
        try{
            System.out.println("Enter number: ");
            Scanner sc = new Scanner(System.in);
            int n = sc.nextInt();
            if(n>0)
            {
                double sqrt1 = Math.sqrt(n);
                System.out.println("Square root is: " + sqrt1);
            }
            else
                throw new ArithmeticException();
        }
        catch (ArithmeticException e)
        {
            System.out.println("Please enter a valid number");
        }
    }
}
```

```
Enter number:
9
Square root is: 3.0

Process finished with exit code 0
```

```
Enter number:
-8
Please enter a valid number

Process finished with exit code 0
```

3.Create a CourseException class that extends Exception and whose constructor receives a String that holds a college course's department (for example, CIS), a course number (for example, 101), and a number of credits (for example, 3). Save the file as CourseException.java. Create a Course class with the same fields and whose constructor requires values for each field. Upon construction, throw a CourseException if the department does not consist of three letters, if the course number does not consist of three digits between 100 and 499 inclusive, or if the credits are less than 0.5 or more than 6. Save the class as Course.java. Write an application that establishes an array of at least six Course objects with valid and invalid values. Display an appropriate message when a Course object is created successfully and when one is not. Save the file as ThrowCourseException.java.

Course.java

```
package Courses;

public class Course
{
    String newLine = System.getProperty("line.separator");
    String department;
    int courseNumber;
    double credits;
    public Course(String department,int courseNumber,double credits)throws
CourseException
```

```
    {
        if(department.length()!=3||(courseNumber<100||
                courseNumber>499)||(credits<0.5|| credits>6))
        {
            try
            {
                throw new CourseException(department,courseNumber,credits);
            }
            catch(CourseException ex)
            {
            }
        }
        else
        {
            System.out.println("Created successfully" + newLine +
"Department name:" +
                    "+department+" +  newLine + "Course number:" +
+courseNumber+  newLine + "credits:" + credits + newLine);

        }
        this.department = department;
        this.courseNumber = courseNumber;
        this.credits = credits;
    }
}
```

CourseException.java

```
package Courses;

public class CourseException extends Exception
{
    public CourseException(String dept,int course,double cred)
    {
        System.out.println("Not created successfully"+"\nDepartment name:
"+dept+"\nCourse number: "+course+"\nCredits: "+cred+"\n");
    }
}
```

ThrowCourseException.java

```
package Courses;

public class ThrowCourseException
{
    public static void main(String[] args) throws CourseException
    {
        Course course1 = new Course("CSE", 101, 3.1);
        Course course2 = new Course("CSE", 2002, 4.0);
        Course course3 = new Course("ECE", 111, 4.0);
        Course course4 = new Course("ECE", 2005, 3.3);
        Course course5 = new Course("CSE", 404, 4.0);
        Course course6 = new Course("CSE",405,13.4);
    }
}
```

```
Created successfully
Department name:+department+
Course number:101
credits:3.1

Not created successfully
Department name: CSE
Course number: 2002
Credits: 4.0

Created successfully
Department name:+department+
Course number:111
credits:4.0

Not created successfully
Department name: ECE
Course number: 2005
Credits: 3.3

Created successfully
Department name:+department+
Course number:404
credits:4.0
```

```
Not created successfully
Department name: CSE
Course number: 405
Credits: 13.4


Process finished with exit code 0
```

4. Create a UsedCarException class that extends Exception; its constructor receives a value for a vehicle identification number (VIN) that is passed to the parent constructor so it can be used in a getMessage() call. Save the class as UsedCarException.java. Create a UsedCar class with fields for VIN, make, year, mileage, and price. The UsedCar constructor throws a UsedCarException when the VIN is not four digits; when the make is not Ford, Honda, Toyota, Chrysler, or Other; when the year is not between 1997 and 2017 inclusive; or either the mileage or price is negative. Save the class as UsedCar.java. Write an application that establishes an array of at least seven UsedCar objects and handles any Exceptions. Display a list of only the UsedCar objects that were constructed successfully. Save the file as ThrowUsedCarException.java.

UsedCar.java

```
package Cars;
import java.lang.*;

public class UsedCar {
    String vin;
    String make;
    int year;
    double mileage;
    double price;
    public UsedCar(String vin, String make, int year, int mileage, int
price) throws UsedCarException {
        if (vin.length()!=4)
            throw new UsedCarException(vin);
        if((!make.equals("Ford"))&& (!make.equals("Honda"))&&
(!make.equals("Toyota"))&&
                (!make.equals("Chrysler"))&& (!make.equals("Other")))
            throw new UsedCarException("Exception: Invalid make");
        if(year<1990||year>2014)
            throw new UsedCarException("Excepetion: Invalid year");
        if(mileage<0)
            throw new UsedCarException("Excepetion: Invalid mileage");
        if(price<0)
            throw new UsedCarException("Excepetion: Invalid price");
        this.vin=vin;
        this.make=make;
        this.year=year;
        this.mileage=mileage;
```

```
        this.price=price;
    }
    public String toString() {
        return "VIN " + vin + " Make: " + make + "\n Year: " + year + " " +
mileage + " miles $" + price;
    }
}
```

UsedCarException.java

```
package Cars;

public class UsedCarException extends Exception {
    public UsedCarException(String s) {
        super("Given information is incorrect.");
    }
}
```

ThrowUsedCarException.java

```
package Cars;

public class ThrowUsedCarException {
    public static void main(String[] args) throws UsedCarException {
        UsedCar[] car = new UsedCar[7];
        try {
            car[0] = new UsedCar("123", "Ford", 1998, 56, 400000);
            System.out.println(car[0].toString() + " is successful");
        }catch (UsedCarException e) {
            System.out.println(e.getMessage());
        }
        try {
            car[1] = new UsedCar("1245", "Honda", 1998, 89, 2600000);
            System.out.println(car[1].toString() + " is successful");
        }catch (UsedCarException e) {
            System.out.println(e.getMessage());
        }
        try {
            car[2] = new UsedCar("12581", "Chrysler", 1995, 77, 9000000);
            System.out.println(car[2].toString() + " is successful");

        }catch(UsedCarException e) {
            System.out.println(e.getMessage());
        }
        try {
            car[3] = new UsedCar("1261", "Mercedes", 1998, -67, 6700000);
            System.out.println(car[3].toString() + " is successful");

        }catch (UsedCarException e){
            System.out.println(e.getMessage());
        }
        try {
            car[4] = new UsedCar("1273", "Toyota", 1998, 45, -700000);
            System.out.println(car[4].toString() + " is successful");
        }catch (UsedCarException e) {
            System.out.println(e.getMessage());
        }
```

```
        try {
            car[5] = new UsedCar("1219", "BMW", 2005, 78, 2000000);
            System.out.println(car[5].toString() + " is successful");
        }catch (UsedCarException e) {
            System.out.println(e.getMessage());
        }
        try {
            car[6] = new UsedCar("1295", "Toyota", 1998, 67, 1000000);
            System.out.println(car[6].toString() + " is successful");
        }catch (UsedCarException e){
            System.out.println(e.getMessage());
        }

    }
}
```

```
Given information is incorrect.
VIN 1245 Make: Honda
 Year: 1998 89.0 miles $2600000.0 is successful
Given information is incorrect.
Given information is incorrect.
Given information is incorrect.
Given information is incorrect.
VIN 1295 Make: Toyota
 Year: 1998 67.0 miles $1000000.0 is successful

Process finished with exit code 0
```