

Chapter 6

Exercise 1

Debug the code below

```
public class BMICalculator;
{
    // Main method
    public static void main(String[] args){
        // declare variables
        double weight = 60;
        double height = 1.70;
        // calculate BMI
        BMI = CalculateBMI(weight,height);
        // print to screen
        System.out.println("Your BMI is " , BMI , ".");
    }

    // method calculating BMI
    public static double calculateBMI(double w,double h){
        double BMI;
        BMI = weight/(height*height);
        return BMI
    }
}
```

Exercise 2

Debug the code below

```
import java.util.Scanner

public class CarProject {
    public static void main(String args []) {
        / create new instance of car
        Car car1 = new Car(BMW,7);

        // ask user how many miles he wants to drive
        Scanner scan = new Scanner(System.in);
        System.out.println("Please enter how many miles you want to drive");
        double miles = scan.next.Double();

        // convert the miles to km
        DistanceConverter distConverter = DistanceConverter;
        int km = distConverter.convert(miles);
    }
}
```

```

        // calculate the liters fuel needed
        double litersFuel = (int) (km/100) * car1.GetFuelConsumption();

        System.out.println("Driving " , miles , " miles with this " ,
            car1.getBrand() , " will consume " , literFuel , " liters
            fuel.");
    }
}

public class Car {
    String brand;
    double fuelConsumption;

    // constructor method
    public Car(String br, double fc){
        br = brand;
        fc = fuelConsumption;
    }
    // methods to return brand and fuelConsumption
    public String getBrand(){
        return brand;
    }
    public double getFuelConsumption(){
        return fuelConsumption;
    }
}

public class DistanceConverter {
    // constructor method
    public DistanceConverter {
    }
    // method to calculate the BMI
    public double convert(int dist_miles) {
        // define conversion factor
        double oneMileInKM = 1.609;
        // conversion
        final double dist_km
        dist_km = dist_miles * oneMileInKM;
    }
}

```

Exercise 3

The code below throws an exception. Add a try/catch block and let the program stop running if an exception is caught.

```

public class Frozen {

    public static void main (String [] args){

        int [] clientid = {1,2,3,4,5};
        String [] clientname = {"Anna", "Elsa", "Olaf", "Hans", "Kristoff"};
        double [] accountbalance = {1000, 5000, 9, 55, 23};

        for(int i = 1; i<=5; i++)
        {
            System.out.println(clientname[i]+" has "+accountbalance[i]+" dollars. \n");
        }
    }
}

```

```
}  
}
```

Exercise 4

Look at the BMI Calculator from Exercise 1. Make the following changes:

- 1) In the calculateBMI method, first ask the user for height and weight. You can use the statements

- **import java.util.Scanner;**
- **Scanner scan = new Scanner(System.in);**
- **System.out.println("Please enter your weight: ");**
- **double weight = scan.nextDouble();**

Make sure to add a try statement to close any resources for input/output being used in case of exceptions occurring. Also, let the calculateBMI method throw an exception if the input is of the wrong type.

- 2) Secondly, in the calculateBMI method, use the inputted values to calculate the BMI. If the inputted values are illegal (for example if they are negative) let the method throw an exception.
- 3) In your main method, call the calculateBMI method and output the result to the screen. Make sure to handle all exceptions that the method could throw using a try/catch block.

Exercise 5

Write an application that uses a for loop to go through each element of the array below.

```
myArray={2,6,8,1,9,0,10,23,7,5,3}
```

Divide 10 by the number and print out the result if the number is even. If the number is odd print "This number is odd".

Exercise 6

The exceptions you catch with a try/catch block are objects of the class Exception and they are not the only things you can catch. Objects of the classes Throwable and Error can also be caught. In fact, Exceptions and Errors are subclasses of the class Throwable. One common Error is the OutOfMemoryError which is thrown

when the memory is full. This can for example happen when an infinite while loop is executed.

Write an application which contains an infinite while loop and use a try/catch block to catch the above-mentioned error. In order to do this, you need to fill the memory, for example by creating too many objects.

Exercise 7

Take a look at the code below. This code seems to be OK at first sight, but is actually quite poorly written when you look closer. Create a main method in which you cause the provided code to do the things listed below. You are only allowed to use the provided code.

- 1) throw the following exceptions
 - a. NullPointerException
 - b. ArrayIndexOutOfBoundsException
 - c. StackOverflowError
- 2) run in an infinite loop

Now improve the class so that this behavior is no longer possible. You are allowed to change constructor, variables and methods, but the class should still offer the same functionality.

```
public class NotSoSmartCity {

    private String[] inhabitantNames;
    private int nbInhabitants;
    private double totalAmountOfMoney;
    private String name;

    public NotSoSmartCity(String name, int nbInhabitants, double
totalAmountOfMoney, String[] inhabitantNames){
        this.name = name;
        this.nbInhabitants = nbInhabitants;
        this.totalAmountOfMoney = totalAmountOfMoney;
        this.inhabitantNames = inhabitantNames;
    }

    public void printCityNameForEachInhabitant(){
        for(int i = nbInhabitants; i!=0;i--){
            System.out.println(this.name);
        }
    }

    public double getAverageAmountOfMoney(){
        return totalAmountOfMoney/nbInhabitants;
    }

    public static String babyNameGenerator(String beginning, int
lettersToAdd){
        if(0 == lettersToAdd){
            return beginning;
        }
        else{
```

```

        return babyNameGenerator(beginning+getRandomLetter(),
lettersToAdd -1);
    }

    private static char getRandomLetter(){
        String alphabet = "abcdefghijklmnopqrstuvwxyz";
        return alphabet.charAt((int)
(Math.random()*alphabet.length()));
    }

    public void printInhabitantNames(){
        for(int i=0; i < nbInhabitants; i++){
            System.out.println(inhabitantNames[i]);
        }
    }

    public static void main(String[] args){
        //Your code here
    }
}

```