Problem 12.1

```
public class problem12_1 {
    public static void main(String[] args) {
        if(args.length !=1) {
            System.out.println("please use format: java Exercise 12_1 operand1
OPERATOR(+ or - or * or /) operand2");
            System.exit(0);
        }
        int result = 0;
        int a,b;

        a=Integer.valueOf(args[0].charAt(0))-48;
        b=Integer.valueOf(args[0].charAt(2))-48;

        try {
            switch (args[0].charAt(1)) {
            case'+':result = a+b;
            break;

            case '-': result = a-b;
            break;

            case'*': result = a*b;
            break;

            case'/': result = a/b;
            break;
            }

if(Character.isDigit(args[0].charAt(0))&&Character.isDigit(args[0].charAt(2))&&
                    (args[0].charAt(1)=='+' || args[0].charAt(1)=='-' ||
args[0].charAt(1)=='*' ||
                    args[0].charAt(1)== '/')) {
                System.out.println(a+" "+ args[0].charAt(1)+" "+b+"=" +result);
            }
            else if(Character.isLetter(args[0].charAt(0))|| args[0].charAt(0)=='+'||
                    args[0].charAt(0)=='-' || args[0].charAt(0)=='*' ||
args[0].charAt(0)=='/') {
                System.out.println("Wrong input: "+args[0].charAt(0));
            }
            else
if(Character.isDigit(args[0].charAt(0))&&Character.isLetter(args[0].charAt(1))) {
                System.out.println("Wrong input:
"+args[0].charAt(0)+""+args[0].charAt(1));
            }
            else
if(Character.isDigit(args[2].charAt(2))&&Character.isLetter(args[0].charAt(1))||Character.isDigi
t(args[0].charAt(1))) {
                System.out.println("Wrong input:
"+args[0].charAt(1)+""+args[0].charAt(2));
            }
            else if(Character.isLetter(args[0].charAt(2))||args[0].charAt(2)=='+'||
args[0].charAt(2)=='-'|| args[0].charAt(2)=='*'|| args[0].charAt(2)=='/') {
                System.out.println("wrong input: "+args[0].charAt(2));
            }
        }
        catch (NumberFormatException nfe) {
            String message = nfe.getMessage();
            System.out.println(message);
        }
    }

}


public class Calc {

    public static void main(String[] args) {
        if(args.length !=3) {
            System.out.println("Usage: java Calcuator operand1 operator operand2");
            System.exit(0);
        }
        int result = 0;

        switch (args[1].charAt(0)) {
        case '+': result = Integer.parseInt(args[0]) +
                Integer.parseInt(args[2]);
        break;
        case '-': result = Integer.parseInt(args[0]) -
                Integer.parseInt(args[2]);
        break;
        case '*': result = Integer.parseInt(args[0]) *
                Integer.parseInt(args[2]);
        break;
        case '/': result = Integer.parseInt(args[0]) /
                Integer.parseInt(args[2]);
        }

        System.out.println(args[0] + ' ' + args[1] +' ' + args[2]
                + " = " + result);

    }

}
```

Problem 12.3

```
import java.util.Random;
import java.util.Scanner;

public class problem12_3 {
    public static void main(String[] args) {
        final int SIZE = 100;

        Random rand = new Random();
        Scanner input = new Scanner(System.in);
        int[] arr = new int[SIZE];

        for(int i=0; i<SIZE;i++) {
            arr[i] = 1+rand.nextInt(100);
        }
        System.out.print("enter the index of the array: ");
        int index = input.nextInt();

        try {
            int value = arr[index];
            System.out.println("corresponding value at index "+index+" is "+value);
        }
```

```
                    catch(ArrayIndexOutOfBoundsException ex) {
                              System.out.println("out of bounds");
                    }

          }

}
```

Problem 12.5

```
public class IllegalTraiangleException extends Exception {
          public IllegalTraiangleException() {
                    System.out.println("sum of two sides is greater than "+"the other side");
          }

}
public class Triangle {
          private double side1;
          private double side2;
          private double side3;

          public Triangle() {
          }
          public Triangle(double s1, double s2, double s3)throws IllegalTraiangleException{
                    side1=s1;
                    side2=s2;
                    side3=s3;
          try {
                    if((side1+side2)<side3||(side2+side3)<side1||(side1+side3)<side2)
                                        throw new IllegalTraiangleException();
          }
          catch(IllegalTraiangleException ex) {
                    System.out.println(ex);
          }

}
public double getside1() {
          return side1;
}
public double getside2() {
        return side2;
}
public double getside3() {
        return side3;
}
public double getarea() {
        double s=(side1+side2+side3)/2;
        double area = s*(s-side1)*(s-side2)*(s-side3);
        return area;
}
public double getperimeter() {
        return (side1+side2+side3);
}
public String toString() {
        return "Triangle: side1= "+side1+" side2= "+side2+" side3="+side3;
 }

}
public class Etest {

        public static void main(String[] args) throws IllegalTraiangleException {
                Triangle triangle= new Triangle(2.0, 3.0, 10.0);

        }

}
```

Problem 12.7

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class problem12_7 {
        public static void main(String[] args) throws IOException{
                try {
                        BufferedReader bf = new BufferedReader (new
InputStreamReader(System.in));
                        System.out.print("enter the binary value: ");
```

```
            String str = bf.readLine();
            int i = binaryToDecimal(str);
            System.out.println("Decimal := "+ i);
        }
        catch(NumberFormatException ex) {
            System.out.println("Number Format Exception occured.program will
exit");
            System.out.println(ex.getMessage());
            System.exit(0);
        }
    }

    public static int binaryToDecimal(String binaryString) {
        int decimalValue = 0;
        for(int i=0; i<binaryString.length(); i++) {
            Character binaryChar = binaryString.charAt(i);
            String tempStr = binaryChar.toString();
            int tempInt = Integer.parseInt(tempStr);
            if(tempInt !=0 && tempInt !=1) {
                System.out.println("Invalid binary number entered. Program will
exit. ");
                System.exit(0);
            }
            decimalValue = decimalValue * 2 + tempInt;
        }
        return decimalValue;
    }

}
```

Problem 12.11

```
import java.io.File;
import java.io.PrintWriter;
import java.util.Scanner;

public class problem12_11 {
    public static void main(String[] args) throws Exception{
        if(args.length !=2) {
            System.out.println("Usage: java Exercise12_11 John Filename");
            System.exit(0);
        }
        File SourceFile = new File(args[1]);
        File temSource = new File("temp.txt");
        if(!SourceFile.exists()) {
            System.out.println("Source file "+args[1]+" does not exist");
            System.exit(0);
        }
        Scanner input = new Scanner(SourceFile);
        PrintWriter output = new PrintWriter(temSource);

        while(input.hasNext()) {
            String s1 = input.nextLine();
            String s2 = s1.replaceAll(args[0], "");
            output.println(s2);
        }
        input.close();
        output.close();

        File newSource = new File("temp.txt");
        File overriteFile = new File(args[1]);
        Scanner newinput = new Scanner(newSource);
        PrintWriter newoutput = new PrintWriter(overriteFile);

        while(newinput.hasNext()) {
            String s1 = newinput.nextLine();
            newoutput.close();
            if(overriteFile.exists()) overriteFile.delete();
        }
    }

}
```

Problem 12.13

```
import java.io.File;
import java.util.Scanner;

public class problem12_13 {
    public static void main(String[] args) throws Exception {
        if(args.length !=1) {
            System.out.println("command-line argument is missing!");
            System.exit(1);
        }
        String fileName = args[0];
        File source = new File(fileName);
        if(!source.exists()) {
            System.out.println(fileName+" file does not exist!");
            System.exit(2);;
        }
        Scanner infile = new Scanner(source);
        String line;

        int characterCount = 0;
        int wordsCount = 0;
        int linesCount = 0;

        while(infile.hasNextLine()) {
            line = infile.nextLine();
            linesCount++;
            String[] words = line.split(" ");
            wordsCount += words.length;

            for(String token: words) {
                characterCount =+ token.length();
            }
        }
        System.out.println("Name of the input file: "+fileName);
        System.out.println("Number of lines in the file: "+ linesCount);
        System.out.println("Number of words in the file: "+ wordsCount);
        System.out.println("Number of characters in the file: "+ characterCount);
    }

}
```