

## HW2 Code – Alon Morad

### 1. Divisors.java

```
/**
 * Gets a command-line argument (int), and prints all the divisors
 * of the given number.
 */
public class Divisors {
    public static void main (String[] args) {
        // declares int variable and gets it from user
        int num = Integer.parseInt(args[0]);
        // runs on all numbers from 1 to num (including num) and
        // checks if i divides num (means num % i equals 0)
        for (int i = 1; i <= num; i++){
            if (num % i == 0)
                // print i if it's a divisor of num
                System.out.println(i);
        }
    }
}
```

## **2. Reverse.java (Reversing a String)**

```
/**
 * Prints a given string, backward. Then prints the middle character
 in the string.
 * The program expects to get one command-line argument: A string.
 */
public class Reverse {
    public static void main (String[] args){
        // declares string variable and gets it from user
        String word = args[0];
        // declares empty string variable
        String reversedWord = "";
        // reversedI == the placement of the last letter, will be used
        for charAt
        int reversedI = word.length()-1;
        // builds the word from end to start decreasing reversedI each
        round of the loop
        for (int i = 0; i < word.length(); i++){
            reversedWord = reversedWord +
            word.charAt(reversedI);
            reversedI--;
        }
        // prints the reversed word
        System.out.println(reversedWord);
        // checks if the number of letters in the word is even - if it
        does, the middle letter will be at (placement / 2) -1
        // if the number of letters in the word is odd the middle
        letter will be at placement / 2
        // prints it depends on the case
        if (word.length() % 2 == 0)
```

```
System.out.println("The middle character is " +  
word.charAt(word.length()/2-1));
```

```
    else
```

```
System.out.println("The middle character is " +  
word.charAt(word.length()/2));
```

```
    }
```

```
}
```

### 3. InOrder.java (Lucky Streak)

```
/**
 * Generates and prints random integers in the range [0,10),
 * as long as they form a non-decreasing sequence.
 */
public class InOrder {
    public static void main (String[] args) {
        // declares and generating first random number and parsing it
        to int
        int firstNum = (int) (Math.random() * 10);
        // prints the first generated number
        System.out.print(firstNum);
        // declares and generating second random number and parsing it
        to int
        int secondNum = (int) (Math.random() * 10);
        // prints the numbers as long as the next generated number is
        smaller than the current number
        while (firstNum <= secondNum){
            System.out.print(" " + secondNum);
            // saves the current number before generating a new one
            firstNum = secondNum;
            // generates a new random number
            secondNum = (int) (Math.random() * 10);
        }
    }
}
```

#### 4. Perfect.java (Perfect Numbers)

```
/**
 * Gets a command-line argument (int), and checks if the given
 * number is perfect.
 */
public class Perfect {
    public static void main (String[] args) {
        // declares int variable and gets it from user
        int num = Integer.parseInt(args[0]);
        // declares int variable divisors and sets his first value as
        1
        int divisors = 1;
        // declares string variable and sets his first value as
        requested sentence
        String perfectNum = num + " is a perfect number since " +
        num + " = 1";
        // runs on all numbers from 2 to num and checks if i divides
        num (means num % i equals 0)
        // if it does it adds i to the string and adds it to sum of
        divisors
        for (int i = 2; i < num; i++){
            if (num % i == 0){
                perfectNum = perfectNum + " + " + i;
                divisors = divisors + i;
            }
        }
        // checks if num equals to sum of divisors
        // if it does it prints the final sentence that was requested
        else it prints that num isn't perfect number
        if (num == divisors)
            System.out.println(perfectNum);
    }
}
```

```
        else
            System.out.println(num + " is not a perfect
number");
    }
}
```

## 5. DamkaBoard.java

```
/**
 * Gets a command-line argument n (int), and prints an n-by-n damka
 * board.
 */
public class DamkaBoard {
    public static void main(String[] args) {
        // declares int variable and gets it from user
        int num = Integer.parseInt(args[0]);
        // run nested for loop: one for rows and one for columns
        // for each even row it creates a little shift
        for (int i = 0; i < num; i++){
            for (int j = 0; j < num; j++){
                if (i % 2 == 0)
                    System.out.print("* ");
                else
                    System.out.print(" *");
            }
            // starts a new line every loop
            System.out.println();
        }
    }
}
```

## 6. OneOfEach.java

```
/**
 * Simulates the formation of a family in which the parents decide
 * to have children until they have at least one child of each
 * gender.
 */
public class OneOfEach {
    public static void main (String[] args) {
        // declares double variable that will be used as a number
        // generator
        double genderNum;

        // declares int variable that will be used as a counter
        int counter = 0;

        // declares two boolean variables that will help determine if
        // boy/girl was already generated
        boolean isBoy = false;
        boolean isGirl = false;

        // runs while loop as long as boy & girl wasn't generated.
        // For values: 0-0.5 it's a boy, for values 0.5-1 it's a girl
        while (isBoy == false | isGirl == false){
            // generates random number using Math.random
            genderNum = Math.random();

            // if the value is under 0.5 == boy, and prints b
            if (genderNum < 0.5){
                isBoy = true;
                System.out.print("b ");
            }

            // if the value is over 0.5 == girl, and prints g
            else{
```



```
        isGirl = true;
        System.out.print("g ");
    }
    // counts every loop to know how many children was born
    counter++;
}
// starts a new line
System.out.println();
// prints the number of total children
System.out.println("You made it... and you now have " +
    counter + " children.");
}
}
```

## 7. OneOfEachStats1.java

```
/**
 * Computes some statistics about families in which the parents
 decide
 * to have children until they have at least one child of each
 gender.
 * The program expects to get one command-line argument: an int
 value
 * that determines how many families to simulate.
 */
public class OneOfEachStats1 {
    public static void main (String[] args) {
        // Gets a command-line argument
        int T = Integer.parseInt(args[0]);

        // declares three int variables that will be used as counters
        for each of the categories
        int counterOf2 = 0;
        int counterOf3 = 0;
        int counerOf4more = 0;

        // declares double variable that will be used to compute the
        average number of children
        double sumOftotalchildren = 0;

        // runs the for loop T times
        for (int i = 0; i < T; i++){
            // declares double variable that will be used as a number
            generator
            double genderNum;

            // declares int variable that will be used as a counter
            int counter = 0;

            // declares two boolean variables that will help determine if
            boy/girl was already generated
```

```

        boolean isBoy = false;
        boolean isGirl = false;
// runs while loop as long as boy & girl wasn't generated.
// For values: 0-0.5 it's a boy, for values 0.5-1 it's a girl
        while (isBoy == false | isGirl == false){
            // generates random number using Math.random
            genderNum = Math.random();
            // if the value is under 0.5 == boy
            if (genderNum < 0.5){
                isBoy = true;
            }
            // if the value is over 0.5 == girl
            else {
                isGirl = true;
            }
        }
// counts every loop to know how many children was born
        counter++;
// sums total number of children
        sumOfTotalChildren++;
    }
// counts the categories using the value of counter
// if the loop ran two times, it adds +1 to countersof2
category
        if (counter == 2)
            counterOf2++;
// if the loop ran three times, it adds +1 to countersof3
category
        else if (counter == 3)
            counterOf3++;
// if the loop ran four times or more, it adds +1 to
countersof4more category

```

```

        else

            counerOf4more++;

        // after checking to which category it was related, resets
        counter for the next loop check

        counter = 0;

    }

    // prints the average number of children

    System.out.println("Average: " + sumOfTotalChildren / T
        + " children to get at least one of each gender.");

    // prints counters of each category

    System.out.println("Number of families with 2 children: "
        + counterOf2);

    System.out.println("Number of families with 3 children: "
        + counterOf3);

    System.out.println("Number of families with 4 or more
        children: " + counerOf4more);

    // check which counter was the maximum and prints it

    if (counterOf2 >= counterOf3 && counterOf2 >=
        counerOf4more)

        System.out.println("The most common number of
            children is 2.");

    else if (counterOf3 >= counterOf2 && counterOf3 >=
        counerOf4more)

        System.out.println("The most common number of
            children is 3.");

    else

        System.out.println("The most common number of
            children is 4 or more.");

    }

}

```

## 8. OneOfEachStats.java

```
import java.util.Random;

/**
 * Computes some statistics about families in which the parents
 * decide
 *
 * to have children until they have at least one child of each
 * gender.
 *
 * The program expects to get two command-line arguments: an int
 * value
 *
 * that determines how many families to simulate, and an int
 * value
 *
 * that serves as the seed of the random numbers generated by the
 * program.
 *
 * Example usage: % java OneOfEachStats 1000 1
 */
public class OneOfEachStats1 {
    public static void main (String[] args) {
        // Gets two command-line arguments
        int T = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);

        // Initailizes a random numbers generator with the given seed
        value
        Random generator = new Random(seed);

        // declares three int variables that will be used as counters
        for each of the categories
        int counterOf2 = 0;
        int counterOf3 = 0;
        int counerOf4more = 0;

        // declares double variable that will be used to compute the
        average number of children
        double sumOftotalchildren = 0;

        // runs the for loop T times
```

```

        for (int i = 0; i < T; i++){
// declares double variable that will be used as a number
generator

            double genderNum;

// declares int variable that will be used as a counter

            int counter = 0;

// declares two boolean variables that will help determine if
boy/girl was already generated

            boolean isBoy = false;

            boolean isGirl = false;

// runs while loop as long as boy & girl wasn't generated.
// For values: 0-0.5 it's a boy, for values 0.5-1 it's a girl
            while (isBoy == false | isGirl == false){
// generates random number using seed & nextDouble

                genderNum = generator.nextDouble();

// if the value is under 0.5 == boy
                if (genderNum < 0.5){

                    isBoy = true;

                }

// if the value is over 0.5 == girl
                else {

                    isGirl = true;

                }

// counts every loop to know how many children was born

                counter++;

// sums total number of children

                sumOftotalchildren++;

            }

// counts the categories using the value of counter
// if the loop ran two times, it adds +1 to countersof2
category

```

```

        if (counter == 2)
            counterOf2++;
        // if the loop ran three times, it adds +1 to countersof3
category
        else if (counter == 3)
            counterOf3++;
        // if the loop ran four times or more, it adds +1 to
countersof4more category
        else
            counerOf4more++;
        // after checking to which category it was related, resets
counter for the next loop check
        counter = 0;
    }
    // prints the average number of children
    System.out.println("Average: " + sumOfTotalChildren / T
        + " children to get at least one of each gender.");
    // prints counters of each category
    System.out.println("Number of families with 2 children: "
        + counterOf2);
    System.out.println("Number of families with 3 children: "
        + counterOf3);
    System.out.println("Number of families with 4 or more
children: " + counerOf4more);
    // check which counter was the maximum and prints it
    if (counterOf2 >= counterOf3 && counterOf2 >=
counerOf4more)
        System.out.println("The most common number of
children is 2.");
    else if (counterOf3 >= counterOf2 && counterOf3 >=
counerOf4more)
        System.out.println("The most common number of
children is 3.");
    else

```

```
System.out.println("The most common number of  
children is 4 or more.");
```

```
}
```

```
}
```