Divisors.java

```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) {
        int x = Integer.parseInt(args[0]);
        for(int i = 1;  i <= x; i++) {
            if(x % i == 0) {
                System.out.println(i);
            }
        }
    }
}
```

Reverse.java

```java
/**
 * 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){
        String str = args[0];
        String reversed = "";
        for(int i = str.length() - 1; i >= 0; i--) {
            reversed += str.charAt(i);
        }
        System.out.println(reversed);
        char midCHar = reversed.charAt((str.length()) / 2);
        System.out.println("The middle character is " + midCHar);
    }
}
```

InOrder.java

```java
/**
 *  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) {
        int randomNumber = (int)(Math.random() * (10));
        int output = 0;

        do {
            output = randomNumber;
            System.out.println(output);
            randomNumber = (int)(Math.random() * (10));
        } while (output < randomNumber);
    }
}
```

Perfect.java

```java
/**
 *  Gets a command-line argument (int), and chekcs if the given number
is perfect.
 */
public class Perfect {
    public static void main (String[] args) {
        int num = Integer.parseInt(args[0]);
        int sum = 1;
        String str = args[0] + " is a perfect number since " + args[0]
+ " = 1";
        for(int i = 2; i < num; i++) {
            if (num % i == 0) {
                sum += i;
                str += " + " + i;
            }
        }
        if(sum == num) {
            System.out.println(str);
        }
        else
            System.out.println(num + " is not a perfect number");
    }
}
```

DamkaBoard.java

```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) {
        int n = Integer.parseInt(args[0]);
        // String output = "";
        for(int i = 0; i < n; i++) {
            System.out.println();
            for(int j = 0; j < n; j++) {
                if((i % 2 == 1))
                    System.out.print(" *");
                else
                    System.out.print("* ");
            }
        }
    }
}
```

OneOfEachStats.java

```
import java.util.Random;
public class OneOfEachStats {
    public static void main (String[] args) {
        // Gets the 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);

        double totalNumOfChildren = 0;
        int twoChildren = 0;
        int threeChildren = 0;
        int fourOrMoreChildren = 0;
        double average = 0;
        String common = "";

        for(int i = 0; i < T; i++){
            boolean boyBorned = false;
            boolean girlBorned = false;
            int children = 0;

            while(!boyBorned || !girlBorned) {
                double rnd = generator.nextDouble();
                if(rnd < 0.5) {
                    boyBorned = true;
                } else {
                    girlBorned = true;
                }
                children++;
                totalNumOfChildren++;
            }

            if(children == 2) {
                twoChildren++;
            } else if(children == 3) {
                threeChildren++;
            } else {
                fourOrMoreChildren++;
            }
        }

        average = totalNumOfChildren / T;
```

```
        if(twoChildren > threeChildren && twoChildren >
fourOrMoreChildren) {
            common = "2";
        } else if(threeChildren > twoChildren && threeChildren >
fourOrMoreChildren) {
            common = "3";
        } else {
            common = "4";
        }

        System.out.println("Average: " + average + " children to get
at least one of each gender.");
        System.out.println("Number of families with 2 children: " +
twoChildren);
        System.out.println("Number of families with 3 children: " +
threeChildren);
        System.out.println("Number of families with 4 or more
children: " + fourOrMoreChildren);
        System.out.println("The most common number of children is " +
common + ".");
    }
}
```