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

2. Reverse.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 s = args[0];
		String newS = "";
		int n= s.length()-1;
		while (n != -1) {
			char c = s.charAt(n);
			newS = newS + c;
			n = n-1;
			}
		char middle;
		if (s.length()%2==0) {
		middle =s.charAt(s.length()/2-1);
		}
		else {
		middle = s.charAt(s.length()/2);
		}
		System.out.println(newS);
		System.out.println("The middle character is " + middle);
	}
}
```

3.InOrder.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) {
                boolean stop = false;
                int rand1 = (int)(Math.random() * 10);
                System.out.println(rand1);
                while (stop != true) {
                        int rand2 = (int)(Math.random() * 10);
                        if (rand2 > rand1) {
                                System.out.println(rand2);
                                rand1=rand2;
                        }
                        else {
                                stop = true;
                        }
                }
        }
}
```

4.perfect.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 n = Integer.parseInt(args[0]);
		String num = args[0];
		String answer = num + " is a perfect number since "+ num + " = 1";
		int sum = 1;
		for (int i = 2; i < n; i++) {
			if (n%i==0) {
			answer = answer + " + "+i;
			sum = sum + i;
			}
		}
	if (sum == n) {
		System.out.println(answer);
	}
	else {
		System.out.println( n + " 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) {
		int n = Integer.parseInt(args[0]);
		for (int i=0 ; i<n ; i++) {
			for (int j=0 ; j<n ; j++) {
				if (i%2 == 0) {
					System.out.print("* ");
				}
				else {
					System.out.print(" *");
				}
			}
			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) {
                boolean boy = false, girl = false;
                int sum = 0;
                String answer = "";
                while (boy == false || girl == false) {
                        double gender = (Math.random());
                        sum = sum + 1;
                                if ( gender > 0.5 ) {
                                        answer = answer + "b ";
                                        boy = true;
                                }
                                else {
                                        girl = true;
                                        answer = answer + "g ";
                                }
                }
                System.out.println (answer);
                System.out.println ("You made it ... and you now have "+ sum + "
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) {
                int t = Integer.parseInt(args[0]);
                int sum = 0, count2 = 0, count3 = 0, count4 = 0;
                double sumall = 0;
                for (int i = 0 ; i < t ; i++) {
                        boolean boy = false, girl = false;
                        sum = 0;
                                while (boy == false || girl == false) {
                                        double gender = (Math.random());
                                        sum++;
                                        if ( gender > 0.5) {
                                                boy = true;
                                        } else {
                                                girl = true;
                                        }
                                }
                        if (sum == 2) {
                                count2++;
                        } else if (sum == 3) {
                                count3++;
                        } else if (sum >= 4) {
                        count4++;
                        }
                        sumall = sumall + sum;
                }
                double average = sumall / (double) t;
                System.out.println ("Average: " + average + " children to get at least
one of each gender.");
                System.out.println ("Number of families with 2 children: "+ count2);
                System.out.println ("Number of families with 3 children: "+ count3);
                System.out.println ("Number of families with 4  or more children: "+
count4);
```

```
                String mostcommon = "2.";
                if ((count3 > count2) && (count3 > count4)) {
                        mostcommon = "3.";
                } else if ((count4 > count2) && (count4 > count2)) {
                                mostcommon = "4 or more.";
                }
        System.out.println ("The most common number of children is "+
mostcommon);
}
}
```

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 OneOfEachStats {
        public static void main (String[] args) {
                int T = Integer.parseInt(args[0]);
                int seed = Integer.parseInt(args[1]);
      Random generator = new Random(seed);
                int sum = 0, count2 = 0, count3 = 0, count4 = 0;
                double sumall = 0;
                for (int i = 0 ; i < T ; i++) {
                        boolean boy = false, girl = false;
                        sum = 0;
                                while (boy == false || girl == false) {
                                        double gender = generator.nextDouble();
                                        sum++;
                                        if ( gender > 0.5) {
                                                boy = true;
                                        } else {
                                                girl = true;
                                        }
                                }
                        if (sum == 2) {
                                count2++;
                        } else if (sum == 3) {
                                count3++;
                        } else if (sum >= 4) {
                        count4++;
                        }
                        sumall = sumall + sum;
                }
                double average = sumall / (double) T;
                System.out.println ("Average: " + average + " children to get at least
one of each gender.");
                System.out.println ("Number of families with 2 children: "+ count2);
                System.out.println ("Number of families with 3 children: "+ count3);
                System.out.println ("Number of families with 4 or more children: "+
count4);
```

```
                String mostcommon = "2.";
                if ((count3 > count2) && (count3 > count4)) {
                        mostcommon = "3.";
                } else if ((count4 > count2) && (count4 > count2)) {
                                mostcommon = "4 or more.";
                }
        System.out.println ("The most common number of children is "+
mostcommon);
        }
}
```