**HW02 Code – CS**

**Divisors.java**

public class Divisors {  
 public static void main (String[] args) {  
 int num = Integer.parseInt(args[0]);  
 //For statement which prints the divisors of the given command line argument using the modulo operator %  
 for (int i = 1; i <= num; i++) {  
 if (num % i == 0) {  
 System.out.println(i);  
 }  
 }  
 }  
}

**Reverse.java**

public class Reverse {  
 public static void main (String[] args){  
 String input = args[0];  
 int length = input.length();  
 int mid = length/2;  
 //Print the reversed string  
 for (int i = length-1; i >= 0 ;i--) {  
 System.out.print(input.charAt(i));  
 }   
 System.out.println();  
 //Print the middle character  
 if (length % 2 == 0) {  
 System.out.println("The middle character is " + input.charAt(mid-1));  
 }  
 else {System.out.println("The middle character is " + input.charAt(mid));}  
 }  
  
}

**InOrder.java**

public class InOrder {  
 public static void main(String[] args) {  
 // Generate the first random number  
 int num = (int) (Math.random() \* 10);  
  
 // Print the first number  
 System.out.print(num);  
  
 // A do-while loop to generate and print the next numbers  
 do {  
 // Generating the next random number  
 int next\_num = (int) (Math.random() \* 10);  
  
 // Checking if the next number is greater or equal to the current number  
 if (next\_num >= num) {  
 //Printing the next number  
 System.out.print(" " + next\_num);  
  
 //Updating the next num  
 num = next\_num;  
 } else {  
 //Getting out of the loop if the sequence isn't a decreasing one  
 break;  
 }  
 } while (true);  
 }  
}

**DamkaBoard.java**

public class DamkaBoard {  
 public static void main(String[] args) {  
 int input = Integer.parseInt(args[0]);  
 int index\_line = 1;  
 System.out.println();  
  
 while(index\_line <= input) {  
 int x = 1;  
 // Printing the damka board using a while loop  
 while (x <= input) {  
 if ((index\_line % 2 == 0)) {  
 //Starting the row with a space if the line is even  
 System.out.print(" \*");   
 } else {  
 System.out.print("\* ");  
 }  
 x = x+1;   
 }   
 System.out.println();  
 index\_line = index\_line + 1;   
 }  
 }  
}

**Perfect.java**

public class Perfect {  
 public static void main(String[] args) {  
   
 // Parse the command-line argument as an integer  
 int num = Integer.parseInt(args[0]);  
 int sum = 1;   
  
 // The string below will represent the divisors string.  
 // We are starting with 1 cause every number is divisible by 1  
   
 String str = "1";  
  
 // Find divisors and add them to the sum, while skipping the trivial divisor 1   
 for (int i = 2; i <= num / 2; i++) {   
 if (num % i == 0) {  
 sum += i;  
 str += " + " + i;  
 }  
 }  
  
 // Check if the sum of divisors equals the original number  
 if (sum == num) {  
 System.out.println(num + " is a perfect number since " + num + " = " + str);  
 } else {  
 System.out.println(num + " is not a perfect number");  
 }  
 }  
}

**OneOfEachStats.java**

import java.util.Random;  
public class OneOfEachStats {  
 public static void main(String[] args) {  
 int experiments = Integer.parseInt(args[0]);  
 int seed = Integer.parseInt(args[1]);  
 Random generator = new Random(seed); //set the seed value  
   
 //Relevent variables for aggregating  
 int two\_children = 0;  
 int three\_children = 0;  
 int four\_or\_more = 0;  
 int total\_experiments = 0;  
   
 //Loop through the specified number of experiments  
 for (int i = 0; i < experiments; i++) {  
 int its\_a\_boy = 0;  
 int its\_a\_girl = 0;  
 int total\_children = 0;  
 while ((its\_a\_boy < 1) || (its\_a\_girl < 1)) {  
 double which\_gender = generator.nextDouble();  
 if (which\_gender < 0.5) {  
 its\_a\_boy++;  
 } else {  
 its\_a\_girl++;  
 }  
 }  
 //Getting the number of total children  
 total\_children = its\_a\_boy + its\_a\_girl;  
  
  
 //Updating counters based on the total number of children  
 if (total\_children == 2) {  
 two\_children++;  
 } else if (total\_children == 3) {  
 three\_children++;  
 } else if (total\_children >= 4) {  
 four\_or\_more++;  
 }  
  
 //Updating the total number of children across all experiments  
 total\_experiments = total\_experiments + total\_children;  
 }  
  
  
 // Display results  
 double averageChildren = (double) total\_experiments / experiments;  
 System.out.println("Average: " + averageChildren + " children to get at least one of each gender.");  
 System.out.println("Number of families with 2 children: " + two\_children);  
 System.out.println("Number of families with 3 children: " + three\_children);  
 System.out.println("Number of families with 4 or more children: " + four\_or\_more);  
  
 // Most common number of children  
 if ((two\_children > three\_children) && (two\_children > four\_or\_more)) {  
 System.out.println("The most common number of children is 2.");  
 } else if ((three\_children > two\_children) && (three\_children > four\_or\_more)) {  
 System.out.println("The most common number of children is 3.");  
 } else if ((four\_or\_more > two\_children) && (four\_or\_more > three\_children)) {  
 System.out.println("The most common number of children is 4 or more.");  
 } else if ((three\_children > two\_children) && (three\_children == four\_or\_more)) {  
 System.out.println("The most common number of children is 3.");  
 } else if ((two\_children > four\_or\_more) && (three\_children == two\_children)) {  
 System.out.println("The most common number of children is 2.");  
 } else if ((two\_children == four\_or\_more) && (three\_children == two\_children)) {  
 System.out.println("The most common number of children is 2.");  
 System.out.println();  
 }   
 }  
}