```java
package project1;

import java.util.*;


/*
@author: Robert Daniels
02/25/2022

Hourly Wage Calculator: prompts user for week worked as an int, and 7 values for hours worked as a double.
It then has logic to calculate pay for that time frame, based on accepted pay practices and bonuses to rate based on
week worked.


*/

public class HourlyWageCalculator{


    public static void main(String[] args){


        Scanner scnr = new Scanner(System.in);
        double[] arrayToProcess;
        char answer = 'y';
        int calcWeek;

        while (answer == 'y'){

            //get weekNumber to pass to getInputFromUser
            calcWeek = getWeekNumber(scnr);

            if (calcWeek == -1){
                System.out.println("You've entered -1 to exit. Goodbye.");
                System.exit(0);
            }

            // get validated hours from user
            arrayToProcess = getInputFromUser(scnr, calcWeek);

            //pass the sanitized user input to output method
            calculatePayForWeek(calcWeek, arrayToProcess);

```

```
            System.out.print("Would you like to calculate pay for another week? y/n: ");
            answer = scnr.next().charAt(0);
            System.out.println("");
        }

        System.out.println("Thanks for using the wage calculator tool.");
    }


    public static int getWeekNumber(Scanner scnr){
            /*
            This method gets passed a scanner from main. It then prompts user for week number.
            The week number is validated if it is an int between 1 and 52. -1 will kill process.
            Continues to loop until killed or validated

            @returns weekNumber as int

            */
            boolean weekNumberSentinel = false;
            int weekNumber = 0;
            String weekNumberError = "Week must be between 1 and 52, please try again to continue.";

            while (weekNumberSentinel != true && weekNumber != -1){
                try {
                    System.out.print("Enter week worked: ");
                    weekNumber = Integer.parseInt(scnr.next());
                }

                catch (NumberFormatException e){
                    System.out.println(weekNumberError);
                    continue;
                }


                if (weekNumber > 0 && weekNumber < 53){
                    weekNumberSentinel = true;
                } else{
                    System.out.println(weekNumberError);
                }
            }

            return weekNumber;

```

```java
    }

    public static double[] getInputFromUser(Scanner scnr, int calcWeek){
        /*
        gets hours worked for the week from user.

        Will continue to loop until validated.

        Hours worked are passed to convertStringToDouble once validated

        @returns outputArray as a double[] once passed back up from convertStringToDouble
        */

        String hoursInput;
        final int DAYS_IN_WEEK = 7;
        String[] inputArray = new String[DAYS_IN_WEEK];
        double[] outputArray = new double[DAYS_IN_WEEK];
        boolean validated = false;

        scnr.nextLine(); //clear the stream

        while (validated != true){
            System.out.printf("Enter hours for week %d: ", calcWeek);
            hoursInput = scnr.nextLine();

            try{
                inputArray = hoursInput.split(" ");
                validated = validateInput(inputArray); // calls validateInput per assignment
            } catch (ArrayIndexOutOfBoundsException e){
                System.out.println("You must enter seven values");
            }

        }

        if (validated == true){
            outputArray = convertStringArrayToDouble(inputArray); //calls convert to double per assignment
        }

        return outputArray;


    }

```

```java
    public static boolean validateInput(String[] inputArray){
        /*
        takes String[] inputArray and checks to see if input is valid.

        Data are validated if:

        Exactly 7 values are given
        Every value must be > 0
        Every value must be able to convert to a double

        @returns validCheck as a boolean

        */

        boolean validCheck7 = false;
        boolean validCheckNumeric = true;
        boolean validCheckPositive = true;
        boolean validCheck15 = true;
        boolean validCheck = false;


        // check for exactly 7 values
        if (inputArray.length == 7){
            validCheck7 = true;
        } else {
            System.out.println("Input must have seven numbers");
        }


        // code from demo to check for all numerics
        for (int i=0; i < inputArray.length; i++){
            try{
                Double.valueOf(inputArray[i]);
            } catch (NumberFormatException e){
                System.out.println("Input must be all numeric values");
                validCheckNumeric = false;
            }
        }


        // check for negative values
        for (int i = 0; i < inputArray.length; i++){
            try{
```

```
173                 if (Double.valueOf(inputArray[i]) > 15){
174                     validCheck15 = false;
175                     System.out.println("Daily hours cannot exceed 15.");
176                 } else if (Double.valueOf(inputArray[i]) >= 0){
177                     continue;
178                 } else{
179                     System.out.println("Input must be all positive values.");
180                     validCheckPositive = false;
181                 }
182             } catch (NumberFormatException e){
183                 continue;
184                 }
185         }
186 
187 
188 
189         //check if all tests pass
190 
191         validCheck = validCheck7 && validCheckNumeric && validCheckPositive && validCheck15;
192 
193         return validCheck;
194     }
195 
196     public static double[] convertStringArrayToDouble(String[] inputArray){
197         /*
198         takes validated inputArray passed from getInputFromUser, loops through array and
199         converts elements to double.
200 
201         @returns outputArray as a double[]
202         */
203 
204 
205         double[] outputArray;
206         outputArray = new double[inputArray.length];
207 
208         for (int i = 0; i < inputArray.length; ++i){
209                 outputArray[i] = Double.valueOf(inputArray[i]);
210         }
211 
212         return outputArray;
213     }
214 
215     public static void calculatePayForWeek(int calcWeek, double[] arrayToProcess){
```

```
    /*
    Using provided pay logic, calculate earned pay for week. Displays output to client
    */

         // declare fixed variables
        final double HOUR_RATE = 15.00;
        final double BONUS_RATE = 2.00;
        final double OVERLOAD_PERCENT = 1.5;
        final double REGULAR_HOURS = 40.0;

        // declare the weeks that would get a bonus
        int[] highDemandWeeks = {1, 2, 44, 45, 46, 47, 48, 49, 50,
                                 51, 52};


        double totalHours = 0;
        double regHoursWorked = 0;
        double overtimeHours = 0;
        int bonusPayFlag = 0; // controls if bonusPay is added to rate or not
        boolean bonusWeek = false;
        double actualRate;
        double totalPay;


        totalHours = getTotalHours(arrayToProcess); // call to getTotalHours per assignment


        // how many hours are overtime, assign regular hours worked
        if (totalHours > 40){
            overtimeHours = totalHours - REGULAR_HOURS;
            regHoursWorked = 40;
        } else{
            regHoursWorked = totalHours;
        }

        //check if a bonus week

       for (int i = 0; i < highDemandWeeks.length; ++i){
            if (calcWeek == highDemandWeeks[i]){
                bonusWeek = true;
                bonusPayFlag = 1;
            }
        }
```

```
259
260
261         actualRate = HOUR_RATE + (BONUS_RATE * bonusPayFlag); // up rate to bonus rate if bonus week
262
263         totalPay = (regHoursWorked * actualRate) + (overtimeHours * (OVERLOAD_PERCENT * actualRate));
264
265        // ****** OUTPUT TO CLIENT *******
266
267        if (bonusWeek == true){
268            System.out.printf("Week %d receives a bonus of $%.2f per hour\n", calcWeek, BONUS_RATE);
269        }
270
271        System.out.printf("Your total pay for week %d is: $%.2f\n", calcWeek, totalPay);
272        System.out.printf("You worked a total of %.2f hours\n", totalHours);
273        System.out.println("Here is your summary: ");
274        System.out.printf("\t Base hours worked: %.2f at $%.2f\n", regHoursWorked, actualRate);
275        System.out.printf("\t Base pay: $%.2f\n", (regHoursWorked * actualRate));
276        System.out.printf("\t Overtime hours worked: %.2f at $%.2f\n", overtimeHours, (actualRate * OVERLOAD_PERCENT));
277        System.out.printf("\t Overtime pay: $%.2f\n", (overtimeHours * (actualRate * OVERLOAD_PERCENT)));
278
279
280         // ****** /OUTPUT TO CLIENT *******
281    }
282
283    public static double getTotalHours(double[] arrayToProcess){ // kept as double, not integer
284        /*
285        Takes the arrayToProcess passed from calculatePay and returns sumTotal hours.
286        Instructions say this should return an integer value, but employees would mutiny if they lost
287        partial hours worked.
288
289        @returns totalHours as a double
290        */
291
292        double totalHours = 0;
293
294        for (int i = 0; i < arrayToProcess.length; ++i){
295            totalHours = totalHours + arrayToProcess[i];
296        }
297
298        return totalHours;
299    }
300}
```