```
/* SELF ASSESSMENT

Harness Class: Member variables (8 marks)
All my data members are declared, private and the ones that don't change are marked as private. I also have a constant for the maximum number
of uses of a harness.
Comment:8        All the members in the classes are marked as private as well as those which aren't meant to change.

Harness Class: Harness constructor 1 & constructor 2 (6 marks)
I initialise all the variables using the parameters given and set the other members to reasonable default values.
Comment:6        I initialised all the variables with the given parameters as well as setting other members to reasonable default values.

Harness Class: checkHarness method (5 marks)
My method takes an instructor's name as a parameter, and if the harness is not on loan sets the instructor member variable to the given
parameter value (assuming the instructor's name is not null/empty). It also resets the number of times the harness was used.
Comment:5        The method take the instructors name and sets all the variable values to the appropriate value.

Harness Class: isHarnessOnLoan method (2 marks)
My method has no parameters and returns the value of the loan status variable.
Comment:2        My program has no parameters and returns a value the appropriate status

Harness Class: canHarnessBeLoaned method (4 marks)
My method has no parameters and returns true if the harness is not on loan and if the number of times it was used is less than the maximum
allowed number of times.
Comment:4        My program has no parameters and returns a value the appropriate status

Harness Class: loanHarness method (6 marks)
My method has a member name as a parameter and it checks if harness can be loaned by using the canHarnessBeLoaned method. If true, it sets the
club member value to the parameter value, sets the on loan status to true and increments the number of times used variable.
Comment:6        My program has the member name as a parameter and checks if harness can be loaned. If so sets the parameter value to the club
member value.

Harness Class: returnHarness method (5 marks)
My method has no parameters, checks if the harness is on loan, and if so, changes its on-loan status to false, and resets the club member
value.
Comment:5        My method has no parameters, checks if the harness is on loan, and if so, changes its on-loan status to false, and resets the
club member value.

Harness Class: toString method (3 marks)
My method returns a String representation of all the member variables.
Comment:3        it returns a string for all the member values

HarnessRecords Class: member variables (3 marks)
I declare the member variable as a private collection of Harnesses
Comment:3        I do declare them as private.

HarnessRecords Class: HarnessRecords constructor 1 & 2 (9 marks)
In the first constructor, I set the member variable to null [1 mark]. In the second constructor, I use the set of characteristics in the list
to create Harness objects and add them to the collection.
Comment:9        I do so for the first constructor and for the second I use the haracteristics in the list to create Harness objects.

HarnessRecords Class: isEmpty method (1 marks)
I return true if the collection is null/empty and, false otherwise.
Comment:1        I do the above.

HarnessRecords Class: addHarness method (5 marks)
My method takes a Harness object as a parameter and adds the harness to the collection.
Comment:5        My program takes a Harness object as a input and adds it to the array.

HarnessRecords Class: findHarness method (6 marks)
My method takes a make and model number as parameters. It checks if a harness with such properties exists and if it does it returns harness
object, otherwise returns null.
Comment:6        It takes the above parameters and then check if a harness with the above properties exist. It does the appropriate answer
afterwards.

HarnessRecords Class: checkHarness method (6 marks)
must take instructor name, make and model number as parameters and return a Harness. If a harness with the make and model number exists by
using the findHarness method and is not on loan, the Harness method checkHarness is called with the instructor name as a parameter and the
updated Harness object is returned. If the harness is not available returns null.
Comment:6        It takes the instructor name, make and model and returns a harness. If they already exist the checkHarness is called and
updated.

HarnessRecords Class: loanHarness method (7 marks)
My method takes a club member name as a parameter and looks for an available harness by calling the method canHarnessBeLoaned be loaned. If an
available harness is found it is loaned by using the Harness method loanHarness with the club member as a parameter, returning the harness. If
there's no available harness null is returned.
Comment:7        My method takes the appropriate parameters and look for an available harness and if it can find one returns it. If it can't
it will return a null value.

HarnessRecords Class: returnHarness method (7 marks)
My method takes a make and model number as parameters. It checks if a harness with those properties exists by using the findHarness method. If
the found harness is not null, it returns the harness object by using Harness method returnHarness, otherwise returns null.
Comment:7        My method takes the appropriate parameters. It then checks if a harness with properties exist and if its on loan. I fso it
returns the harness.

HarnessRecords Class: removeHarness method (8 marks)
My method takes a make and model number as parameters and check the collection for a harness with those properties and removes it. It returns
```

```java
the harness object if it is found, otherwise returns null.
Comment:8        It takes the make and model number and checks the collection for a harness with the properties and removes it. Otherwise
returns null.

GUI (Java main line) (5 marks)
My test class has a menu which implements at least the five points specified using the HarnessRecords class methods.
Comment:5        My GUI menu implements 5 points speccified in the question.

*/



import java.util.Scanner;

public class ClimbingClubHarnessRecords {

        public static void main(String[] args) {

                Scanner input = new Scanner(System.in);
                int number=0;

                boolean finished=false;
                HarnessRecords harnessRecords = new HarnessRecords();
                while(!finished)
                {
                        System.out.println("What do you wish to do \n");
                        System.out.println("(1) add a record for a newly purchased harness."
                                        +"\n(2) remove a climbing harness from the club."
                                        +"\n(3) record the fact that a club instructor has checked the safety of a harness."
                                        +"\n(4) loan a harness to a club member if there is an availabe harness."
                                        + "\n(5) return a harness which is no longer in use by a club member."
                                        + "\nPlease type the number corrosponding to what you want");
                        if(input.hasNextInt())
                        {
                                number=input.nextInt();
                        }
                        else
                        {
                                input.next();
                                number= 6;
                        }
                        switch (number)
                        {
                        case 1:
                                System.out.println("What is the name of the last instructor to test it?");
                                String lastInstructorToTestIt=input.next();
                                System.out.println("What is the make of the harness?");
                                String makeOfHarness=input.next();
                                System.out.println("What is the model number of the new harness?");
                                if(input.hasNextInt())
                                {
                                        int modelNumber = input.nextInt();
                                        Harness newHarness = new Harness(makeOfHarness, modelNumber, lastInstructorToTestIt);
                                        harnessRecords.addHarness(newHarness);
                                }
                                else
                                {
                                        input.next();
                                        System.out.println("What you have inputed does not classify as a number. The harness has not been
added to the records. Please try again.\n\n");
                                }
                                break;
                        case 2:
                                System.out.println("What harness do you wish to remove?");
                                if(input.hasNextInt())
                                {
                                        int harnessToRemove=input.nextInt() - 1;
                                        harnessRecords.removeHarness(harnessToRemove);
                                }
                                else
                                {
                                        input.next();
                                        System.out.println("What you have inputed does not classify as a number. The harness has not been
removed. Please try again.\n\n");
                                }
                                break;
                        case 3:
                                System.out.println("What is the name of the last instructor to test it?");
                                String lastInstructorToTest=input.next();
                                System.out.println("What is the make of the harness?");
                                String makeOfTheHarness=input.next();
                                System.out.println("What is the model number of the new harness?");
                                if(input.hasNextInt())
                                {
                                        int modelNumber = input.nextInt();
                                        harnessRecords.checkHarness(lastInstructorToTest, makeOfTheHarness,
modelNumber);
```

```java
                                }
                                else
                                {
                                        input.next();
                                        System.out.println("The harness could not be found. Please try again.\n\n");
                                }
                                break;
                        case 4:
                                System.out.println("What is the name of the climbing club member?");
                                String climbingClubMember=input.next();
                                harnessRecords.loadHarness(climbingClubMember);
                                break;
                        case 5:
                                System.out.println("What is the make of the harness?");
                                String harnessMake=input.next();
                                System.out.println("What is the model number of the new harness?");
                                if(input.hasNextInt())
                                {
                                        int modelNumber = input.nextInt();
                                        harnessRecords.returnHarness(harnessMake, modelNumber);
                                }
                                else
                                {
                                        input.next();
                                        System.out.println("The harness could not be found. Please try again.\n\n");
                                }
                                break;
                        default:
                                System.out.println("What you have inputed is not an acceptable answer. Please try again.\n\n");
                                break;
                        }
                }
                input.close();

        }

}
```