

ICS Assignment submission by CT0354307 Gabriel D'Cotta

Incase any of the code breaks, my code is also on my github: [GITHUB repo](https://github.com/PcVOXels/ics-java-ass)

Or: <https://github.com/PcVOXels/ics-java-ass>

Not sure why office keeps removing my formmating 🤖 so screw it, these are here just in case.

- Question 1

A) Defining problem

Input(s)	Processing	Output(s)
- dHeight - dWeight	- Prompt for "height" and "weight" - Get "height" and "weight" - Calculate BMI - $BMI = \text{weight}(\text{kg}) / (\text{height}(\text{m})^2)$ - Display "dBMI" and status	- dBMI - status

B) Pseudo code

```
//=====
1,    Prompt for "dHeight" in meters and "dWeight" in kilograms
2,    Get "dHeight" and "dWeight"
3,    // Simple validity check
      IF (dHeight < 0.5 OR dWeight < 10) THEN
          DISPLAY "invalid height OR weight"
      ELSE
          dBMI = dWeight/(dheight^2)
      ENDIF
4,    // DISPLAY dBMI and status
      IF (dBMI <= 24) THEN
          DISPLAY dBMI and status = "underweight"
      ELSEIF (dBMI > 24 AND dBMI <= 29) THEN
          DISPLAY dBMI and status = "healthy"
```

```

ELSEIF (dBMI > 29 AND dBMI <= 34) THEN
    DISPLAY dBMI and status = "overweight"
ELSE IF (dBMI > 45) THEN
    DISPLAY dBMI and status = "obese"
ENDIF
ENDIF
ENDIF
ENDIF
END

```

//=====

C) Desk checking

i. Input data

-	Data set 1	Data set 2	Data set 3
dHeight	1.73	1.58	0.3
dWeight	54.0	60.0	8.0

ii. Expected results

-	Data set 1	Data set 2	Data set 3
dBMI	18.042	24.034	-
status	Underweight	healthy	-
message	Your BMI is 18.04 and your status is: Underweight.	Your BMI is 24.03 and your status is: Healthy.	Invalid height or weight.

iii. Desk checking table

Statement number	height	weight	BMI	Message
First Pass				

1,2	1.73	54.0		
3			18.042	
4			DISPLAY	Your BMI is 18.04 and your status is: Underweight.
Second Pass				
1,2	1.58	60.0		
3			24.034	
4			DISPLAY	Your BMI is 18.04 and your status is: Underweight.
Third Pass				
1,2	0.3	8.0		
3				Invalid height or weight.
4				

iv. Pictures of tests

i. First pass

```
Enter your height in meters > 1.73
Enter your weight in kilograms > 54.0
Your BMI is 18.04, and your status is: Under weight.
```

ii. Second pass

```
Enter your height in meters > 1.58
Enter your weight in kilograms > 60.0
Your BMI is 24.03, and your status is: Healthy.
```

iii. Third pass (Invalid height and weight test, had to input a valid input to continue)

```
Enter your height in meters > 0.3
Please enter a valid height in meters and more than 0.5m > 1.0
Enter your weight in kilograms > 8.0
Please enter a valid weight in KG and more than 10kg > 10
Your BMI is 10.00, and your status is: Under weight.
```

iv. Invalid input tests

```
Enter your height in meters > 1--1
Please enter a valid height in meters and more than 0.5m > ..
Please enter a valid height in meters and more than 0.5m > --
Please enter a valid height in meters and more than 0.5m > !@#$
Please enter a valid height in meters and more than 0.5m > asdas
Please enter a valid height in meters and more than 0.5m > 123asd
Please enter a valid height in meters and more than 0.5m > 123.123.123
Please enter a valid height in meters and more than 0.5m > -123
Please enter a valid height in meters and more than 0.5m > 1.73
Enter your weight in kilograms > 123asd
Please enter a valid weight in KG and more than 10kg > asd
Please enter a valid weight in KG and more than 10kg > 123.123.123
Please enter a valid weight in KG and more than 10kg > 54.0
Your BMI is 18.04, and your status is: Under weight.
```

D) Actual code

```
//=====
```

```
import java.util.Scanner;
```

```
public class Qn1Test {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        int iDotCount = 0;
```

```
        String sInput = " ";
```

```
        double dHeight = 0.0, dWeight = 0.0, dBMI = 0.0;
```

```
        Boolean blsNum = true, bHasInput = false;
```

```
        System.out.print("Enter your height in meters > ");
```

```
        sInput = input.nextLine();
```

```
        LABELtestForNum: while (!bHasInput) {
```

```
            LABELinner: while (blsNum) {
```

```
                for (int i = 0; i < sInput.length(); i++) {
```

```
                    Boolean bFlag = Character.isDigit(sInput.charAt(i));
```

```
    if (bFlag) {  
        blsNum = true;  
    } else if (sInput.charAt(i) == '.') {  
        iDotCount++;  
  
        if (i == 0) {  
            break LABELinner;  
        } else if (iDotCount > 1) {  
            break LABELinner;  
        } else {  
            blsNum = true;  
        }  
    } else {  
        blsNum = false;  
        break LABELinner;  
    }  
}  
  
iDotCount = 0;  
  
dHeight = Double.parseDouble(sInput);  
  
if (dHeight < 0.5) {  
    break LABELinner;  
} else {  
    break LABELtestForNum;  
}
```

```
}
```

```
System.out.print("Please enter a valid height in meters and more than 0.5m > ");
```

```
sInput = input.nextLine();
```

```
bIsNum = true;
```

```
iDotCount = 0;
```

```
}
```

```
// System.out.println("> " + dHeight); // test check print out
```

```
System.out.print("Enter your weight in kilograms > ");
```

```
sInput = input.nextLine();
```

```
LABELtestForNum: while (!bHasInput) {
```

```
    LABELinner: while (bIsNum) {
```

```
        for (int i = 0; i < sInput.length(); i++) {
```

```
            Boolean bFlag = Character.isDigit(sInput.charAt(i));
```

```
            if (bFlag) {
```

```
                bIsNum = true;
```

```
            } else if (sInput.charAt(i) == '.') {
```

```
                iDotCount++;
```

```
            if (i == 0) {
```

```
                break LABELinner;
```

```
            } else if (iDotCount > 1) {
```

```
                break LABELinner;
```

```
            } else {
```

```

        blsNum = true;
    }
} else {
    blsNum = false;
    break LABELinner;
}
}

iDotCount = 0;

dWeight = Double.parseDouble(sInput);

if (dWeight < 10) {
    break LABELinner;
} else {
    break LABELtestForNum;
}
}

System.out.print("Please enter a valid weight in KG and more than 10kg > ");
sInput = input.nextLine();
blsNum = true;
iDotCount = 0;
}

// System.out.println("> " + dWeight); // test check print out

```

```

    dBMI = dWeight / (dHeight * dHeight);

    if (dBMI <= 24) {
        System.out.printf("Your BMI is %.2f, and your status is: Under weight.\n", dBMI);
    } else if (dBMI > 24 && dBMI <= 29) {
        System.out.printf("Your BMI is %.2f, and your status is: Healthy.\n", dBMI);
    } else if (dBMI > 29 && dBMI <= 34) {
        System.out.printf("Your BMI is %.2f, and your status is: Over weight.\n", dBMI);
    } else if (dBMI > 34) {
        System.out.printf("Your BMI is %.2f, and your status is: Obese and thicc.\n",
dBMI);
    }
}
}
}

//=====

```

E) Code documentation and explanation

The way my program works is by first taking the user's input as a string in the next line to check the whole input. This is to check for alphabets, symbols and spaces per character in the string variable. The reason for storing the input as a string is that, if an input contains any characters, it will crash the program if the variable type is a double or integer as those characters cannot be stored as those variable types. Hence, the input is a string to store the input and to check for these illegal characters with an alphanumeric checker (which took way too long to write up – 4 hours).

How the alphanumeric checker works is via a for loop (which is inside a while loop with a condition that the input is still a number after each character that is checked and is not an illegal character), where the maximum value of "i" is less than the input length. Each character from left to right is passed through a boolean to check if it is a digit. If the current character that is being checked is not a digit, it will break out from the loop and set the boolean for the input being a number to false. Then reprompts the user to reenter a new and valid digit.

However, numbers have decimal points and negative numbers, so we have to create a special check list for these cases. For negative numbers, we just need to check if the value of "i" in the for loop is 0. if not, if the character at i is '-' (as this means that the input is -> 123-123, where

the '-' is not the first character, or -123 where the first character is a '-' but the second character is also a '-' and is therefore not a valid number), we break from the loop and reprompt for a new value. For decimal numbers, we first check if it's the first character at "i" is 0, if it is, we break and reprompt for a new value. If its not the first character, we add a dot counter to check how many dots the string contains. If it's the first one, the dot check will be 1 and its still a valid number. However, if the string contains a second dot, when the counter for the amount of dots is more than 1, it will break from the loop and reprompt the user for a new number.

I have tried my best to account for as many special cases to return as error casses, but there may be more that I cannot think of. For the most part, the alphanumeric checker works fine for what I need it to do.

Else, the check for height and weight are the same. If a number that is too small to be real (used google to deteremine the world's shortest and lightest people in the world) is inputted by the user, they will be asked to enter a new number that is valid.

Once all validity checks are done and the inputs are valid, the BMI will then be calculated with the formula of $(BMI = weight / (height^2))$. The status (underweight, healthy, overweight and obese) and BMI based on the height and weight inputs from the user are then printed to the output/terminal window.

- Question 2

A) Defining problem

Input(s)	Processing	Output(s)
- dLvl	- prompt for rainfall level "dLvl" - get "dLvl" - compare "dLvl" to "dLvlMin" and "dLvlMax" - calculate the "dLvlAvg" - iEntryCount++ if the input is valid - display "iEntryCount", "dLvl", "dLvlMin", "dLvlMax" and "dLvlAvg"	- iEntryCount - dLvl - dLvlMin - dLvlMax - dLvlAvg

B) Pseudo code

```
//=====
```

- 1, Prompt for rainfall level "dLvl"
- 2, Get "dLvl"
- 3, SET "dLvlMin" = "dLvl", SET "dLvlMax" = "dLvl", SET "iEntryCount" = 0
- 4, // check for "dLvl" value

```

        DOWHILE (dLvl > 0)

            iEntryCount++                // init to 0 before while loop

        IF (dLvl < dLvlMin) THEN

            dLvlMin = dLvl;              // init before while loop to first "dLvl"

        ENDIF

        IF (dLvl > dLvlMax) THEN

            dLvlMax = dLvl;              // init before while loop to first "dLvl"

        ENDIF

        dTotal += dLvl;                  // init to 0 before while loop
5,            dLvlAvg = dTotal / iEntryCount;
6,            Display "iEntryCount", "dLvl", "dLvlMin", "dLvlMax" and "dLvlAvg"
-            DOWHILE (bNoInput)
7,                Prompt new "dLvl"
8,                Get new "dLvl"
9,                IF (dLvl == 0) THEN

                    iEntryCount++, "dLvlMin" = 0;

                                dTotal += dLvl;                // init to 0 before while loop

                    dLvlAvg = dTotal / iEntryCount;
10,                Display "iEntryCount", "dLvl", "dLvlMin", "dLvlMax" and
                    "dLvlAvg"

                END

11,                ELSEIF (dLvl < 0) THEN

                    Display program terminated

                END

12,                ELSEIF (dLvl = NaN)

                    bNoInput = true

        DISPLAY enter a valid input

```

```

13,                ELSE
bNoInput = false
                ENDIF
                ENDIF
                ENDIF
            ENDO
        ENDDO
    END

//=====

```

C) Desk checking

i. Input data

-	Data set 1	Data set 2	Data set 3
dLvl	10	15	2.0
dLvl	5	2	asd
dLvl	0	-1	-10

ii. Expected results

-	Data set 1	Data set 2	Data set 3
dLvlMin	0	2	2
dLvlMax	10	15	2
dLvlAvg	5	8.5	2
iEntryCount	3	2	1
Message	Min: 0 Max: 10 Average: 5 Total entries: 3	Min: 2 Max: 15 Average: 8.5 Total entries: 2	Min: 2 Max: 2 Average: 2 Total entries: 1

iii. Desk checking table

Statement number	dLvl	iEntryCount	dLvlMin	dLvlMax	dLvlAvg	Message
First Pass						
1,2	10	0				
3			10	10		
4		1				
5					10	
6	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	Min: 10 Max: 10 Average: 10 Total entries: 1
7,8	5					
4		2	5	10		
5					7.5	
6	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	Min: 5 Max: 10 Average: 7.5 Total entries: 2
7,8	0					
9		3	0	10	5	
10	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	Min: 0 Max: 10 Average: 5 Total entries: 3 End of program
-	-	-	-	-	-	-
Second Pass						
1,2	15	0				
3			15	15		
4		1				

5					15	
6	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	Min: 15 Max: 15 Average: 15 Total entries: 1
7,8	2					
4		2	2	15		
5					8.5	
6	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	Min: 2 Max: 15 Average: 8.5 Total entries: 2
7,8	-1					
11		DISPLAY				Program terminated with 2 entries
-	-	-	-	-	-	-
Third Pass						
1,2	2.0	0				
3			2	2		
4		1				
5					2	
6	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	Min: 2 Max: 2 Average: 2 Total entries: 1
7,8	asd					
12						Please enter a valid input
7,8	-10					
11						Program terminated with 1 entries

iv. Pictures of test

i. First pass

```
Enter level > 10
1) Rainfall level = 10.00
   Minimum level = 10.00
   Maximum level = 10.00
   Average level = 10.00
Enter level > 5
2) Rainfall level = 5.00
   Minimum level = 5.00
   Maximum level = 10.00
   Average level = 7.50
Enter level > 0
3) Rainfall level = 0.00
   Minimum level = 0.00
   Maximum level = 10.00
   Average level = 5.00
==Ended with 3 entries==
```

ii. Second pass

```
Enter level > 15
1) Rainfall level = 15.00
   Minimum level = 15.00
   Maximum level = 15.00
   Average level = 15.00
Enter level > 2
2) Rainfall level = 2.00
   Minimum level = 2.00
   Maximum level = 15.00
   Average level = 8.50
Enter level > -1
==Terminated with 2 entries==
```

iii. Third pass (Invalid rain level test, had to input a valid input to continue)

```
Enter level > 2.0
1) Rainfall level = 2.00
   Minimum level = 2.00
   Maximum level = 2.00
   Average level = 2.00
Enter level > asd
Please enter a number > -10
==Terminated with 1 entries==
```

iv. Invalid input tests

```
Enter level > 1--1
Please enter a number > ..
Please enter a number > --
Please enter a number > !@#$
Please enter a number > asdas
Please enter a number > 123asdasd
Please enter a number > 123.123.123
Please enter a number > -123
==Terminated with 0 entries==
```

D) Actual code

```
//=====
```

```
import java.util.Scanner;
```

```
public class Qn2Test {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        double dLvl = 1.0, dLvlMin = 0.0, dLvlMax = 0.0, dLvlAvg = 0.0, dTotalLvl = 0.0;
```

```
        int iEntryCount = 0, iDotCount = 0;
```

```
        String sLvl;
```

```
        Boolean blsNum = true, bHasInput = false;
```

```
        LABELrainLevel: while (dLvl >= 0) {
```

```
            System.out.print("Enter level > ");
```

```
            sLvl = input.nextLine();
```

```
            // NaN check
```

```
            LABELtestForNum: while (!bHasInput) {
```

```
                LABELinner: while (blsNum) {
```

```

for (int i = 0; i < sLvl.length(); i++) {
    Boolean bFlag = Character.isDigit(sLvl.charAt(i));
    if (bFlag) {
        blsNum = true;
    } else if (sLvl.charAt(i) == '.') {
        // if its a '.' on i, check if the prev is '-' or '.', if true, break
        iDotCount++;

        if (i == 0) {
            blsNum = true;
        } else if (sLvl.charAt(i - 1) == '-' || sLvl.charAt(i - 1) == '.') {
            break LABELinner;
        } else if (iDotCount > 1) {
            break LABELinner;
        } else {
            blsNum = true;
        }
    } else if (sLvl.charAt(i) == '-') {
        if (i == 0) {
            blsNum = true;
        } else if (sLvl.charAt(i - 1) == '-' || sLvl.charAt(i - 1) == '.'
            || (sLvl.charAt(i - 1) >= 48 && sLvl.charAt(i - 1) <= 57)) {
            // if char at i-1 == '-' || '.' || 0-9, when char == '-', break out
            break LABELinner;
        } else {
            blsNum = true;
        }
    }
}

```



```
    } else if (sLvl.charAt(i) == ' ') {  
        blsNum = false;  
        break LABELinner;  
    } else {  
        blsNum = false;  
        break LABELinner;  
    }  
}
```

```
iDotCount = 0;
```

```
dLvl = Double.parseDouble(sLvl);  
break LABELtestForNum;  
}
```

```
// at blsNum == false, comes out here  
// prompt new input  
System.out.print("Please enter a number > ");  
sLvl = input.nextLine();  
blsNum = true;  
iDotCount = 0;  
}
```

```
if (iEntryCount == 0) {  
    dLvlMin = dLvl;  
    dLvlMax = dLvl;  
}
```

```

    if (dLvl < 0) {
        System.out.printf("==Terminated with %d entries==\n", iEntryCount);

        break LABELrainLevel;
    }

    iEntryCount++;
    dTotalLvl += dLvl;
    dLvlAvg = dTotalLvl / iEntryCount;

    if (dLvl < dLvlMin) {
        dLvlMin = dLvl;
    }
    if (dLvl > dLvlMax) {
        dLvlMax = dLvl;
    }

    System.out.printf(
        "%d) Rainfall level = %.2f\n  Minimum level = %.2f\n  Maximum level = %.2f\n  Average level = %.2f\n",
        iEntryCount, dLvl, dLvlMin, dLvlMax, dLvlAvg);

    if (dLvl == 0) {
        System.out.printf("==Ended with %d entries==\n", iEntryCount);
        break LABELrainLevel;
    }

```

```

    }
}
}
//=====

```

E) Code documentation and explanation

The way my program works is by first taking the user's input as a string in the next line to check the whole input. This is to check for alphabets, symbols and spaces per character in the string variable. The reason for storing the input as a string is that, if an input contains any characters, it will crash the program if the variable type is a double or integer as those characters cannot be stored as those variable types. Hence, the input is a string to store the input and to check for these illegal characters with an alphanumeric checker.

The alphanumeric checker works the same as question 1 which is via a for loop (which is inside a while loop with a condition that the input is still a number after each character that is checked and is not an illegal character), where the maximum value of "i" is less than the input length. Each character from left to right is passed through a boolean to check if it is a digit. If the current character that is being checked is not a digit, it will break out from the loop and set the boolean for the input being a number to false. Then reprompts the user to reenter a new and valid digit.

However, numbers have decimal points and negative numbers, so we have to create a special check list for these cases. For negative numbers, we just need to check if the value of "i" in the for loop is 0. if not, if the character at i is '-' (as this means that the input is -> 123-123, where the '-' is not the first character, or -123 where the first character is a '-' but the second character is also a '-' and is therefore not a valid number), we break from the loop and reprompt for a new value. For decimal numbers, we first check if it's the first character at "i" is 0, if it is, we break and reprompt for a new value. If its not the first character, we add a dot counter to check how many dots the string contains. If it's the first one, the dot check will be 1 and its still a valid number. However, if the string contains a second dot, when the counter for the amount of dots is more than 1, it will break from the loop and reprompt the user for a new number.

I have tried my best to account for as many special cases to return as error casses, but there may be more that I cannot think of. For the most part, the alphanumeric checker works fine for what I need it to do.

Else, the check for the rain level is pretty straight forward. Once all validity checks are done and the inputs are valid, The first input value (dLvl) will be stored to the minimum and maximum rainfall level. If the input value is more than 0, it will compare the value to the minimum and

maximum rainfall and update accordingly. Where if the current value is lower than the minimum, it will be set as the new minimum. And if the current value is higher than the maximum, the new new value will be set as the new maximum.

For each valid number that is 0 or more, we add 1 to the entry counter (iEntryCount++). This is done not only to show how many inputs we have collected, but also to calculate the average rainfall level by adding the current number to the total rainfall level then dividing them by the amount of entries (dTotalLvl += dLvl) and (dLvlAvg = dTotalLvl / iEntryCount).

Once we calculate the average, we then print out the current rainfall level, the minimum, the maximum, and the average rainfall level from the inputs. If however, the value given is less than 0 (negative number), it will end the program and print the total amount of entries before termination.

- Question 2

A) Defining problem

Input(s)	Processing	Output(s)
- dHRrest - dAge - cGender	- prompt for "dHRrest", "dAge" and "cGender" - get "dHRrest", "dAge" and "cGender" - calculate VO2Max -> - dHRmax = 220 – dAge - dVOMax = 15 * (dHRmax / dHRrest) - determine which sport the user is eligible based on their VO2max with IF statements - iSportCount++ if eligible for a sport	- iSportCount - message with the eligible sports

B) Pseudo code

```
//=====
```

- 1, Prompt for "dHRrest", "dAge" and "cGender"
- 2, Get "dHRrest", "dAge" and "cGender"
- 3, SET dHRmax = 220 – dAge;
- 4, SET dVOMax = 15 * (dHRmax / dHRrest)

```
IF (dAge >= 18 && dAge <= 30) THEN
```

```
  IF (cGender == 'm' AND (dVOMax >= 40 AND dVOMax <= 60)) THEN
```

```

5,                iSportCount++
Display mens basketball
ELSEIF (Gender == 'f' AND (dVOMax >= 43 AND dVOMax <= 60) THEN
6,                iSportCount++
Display womens basketball
ENDIF
ENDIF
ENDIF
IF (dAge >= 18 && dAge <= 26) THEN
IF (cGender == 'm' AND (dVOMax >= 62 AND dVOMax <= 74)) THEN
7,                iSportCount++
                Display mens cycling
                ELSEIF (Gender == 'f' AND (dVOMax >= 47 AND dVOMax <= 57) THEN
8,                iSportCount++
                Display womens cycling
ENDIF
ENDIF
IF (cGender == 'm' AND (dVOMax >= 55 AND dVOMax <= 67)) THEN
9,                iSportCount++
                Display mens canoeing
                ELSEIF (Gender == 'f' AND (dVOMax >= 47 AND dVOMax <= 67) THEN
10,               iSportCount++
                Display womens canoeing
ENDIF
ENDIF
ENDIF
IF (dAge >= 18 && dAge <= 22) THEN

```

```

IF (cGender == 'm' AND (dVOMax >= 52 AND dVOMax <= 58)) THEN
11,          iSportCount++
              Display mens gymnastics
              ELSEIF (Gender == 'f' AND (dVOMax >= 36 AND dVOMax <= 50) THEN
12,          iSportCount++
              Display womens gymnastics

ENDIF
ENDIF
ENDIF
IF (dAge >= 10 && dAge <= 25) THEN
IF (cGender == 'm' AND (dVOMax >= 50 AND dVOMax <= 70)) THEN
13,          iSportCount++
              Display mens swimming
              ELSEIF (Gender == 'f' AND (dVOMax >= 40 AND dVOMax <= 60) THEN
14,          iSportCount++
              Display womens swimming

ENDIF
ENDIF
ENDIF
IF (iSportCount == 0) THEN
15,          Display you don't qualify for any sport
              ELSEIF (iSportCount == 1) THEN
16,          Display you qualify for 1 sport
              ELSEIF (iSportCount > 0) THEN
17,          Display you qualify for "iSportCount" sports

ENDIF
ENDIF

```

ENDIF

END

//=====

C) Desk checking

i. Input data

-	Data set 1	Data set 2	Data set 3
dHRrest	90	60	40
dAge	24	18	8
cGender	f	m	b

ii. Expected results

-	Data set 1	Data set 2	Data set 3
dVOmax	32.666	50.5	79.5
iSportCount	0	2	0
Message	You do not qualify for any sport	You qualify for 2 sports Mens basketball Mens swimming	Please enter a valid gender Too young to take part in any sport.

iii. Desk checking table

Statement number	dHRrest	dAge	cGender	dVOmax	iSportCount	Message
First Pass						
1,2	90	24	f		0	
3,4				32.666		
15					Display	You do not qualify for any sport
-	-	-	-	-	-	-
Second pass						

1,2	60	18	m			
3,4				50.5		
5					1	Mens basketball
13					2	Mens swimming
17					Display	You qualify for 2 sports
-	-	-	-	-	-	-
Third pass						
1,2	40	8	b			Please enter a valid gender You are too young to take part in any sport
3,4				79.5		

iv. Pictures of test

i. First pass

```
Enter your heart rate at rest > 90
Enter your age > 24
Enter your gender [m/f] > f
VO2Max > 32.666666666666664
You do not qualify for any type of sport.
```

ii. Second pass

```
Enter your heart rate at rest > 60
Enter your age > 18
Enter your gender [m/f] > m
VO2Max > 50.5
Mens Basketball
Mens Swimming
You qualify for 2 sports.
```

iii. Third pass (Invalid rain level test, had to input a valid input to continue)


```
Enter your heart rate at rest > 40
Enter your age > 8
Enter your gender [m/f] > b
Please enter a valid gender [m/f] > m
VO2Max > 79.5
You're too young to take part in any sport.
You do not qualify for any type of sport.
```

D) Actual code

```
//=====
```

```
import java.util.Scanner;
```

```
public class Qn3Test {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        int iDotCount = 0, iSportCount = 0;
```

```
        char cGender = ' ';
```

```
        double dHRrest = 0.0, dHRmax = 0.0, dAge = 0.0, dVOmax = 0.0;
```

```
        String sInput = " ";
```

```
        Boolean blsNum = true, blsChar = true, bHasInput = false;
```

```
        System.out.print("Enter your heart rate at rest > ");
```

```
        sInput = input.nextLine();
```

```
        LABELtestForNum: while (!bHasInput) {
```

```
            LABELinner: while (blsNum) {
```

```
                for (int i = 0; i < sInput.length(); i++) {
```

```
                    Boolean bFlag = Character.isDigit(sInput.charAt(i));
```

```
    if (bFlag) {  
        blsNum = true;  
    } else if (sInput.charAt(i) == '.') {  
        iDotCount++;  
  
        if (i == 0) {  
            break LABELinner;  
        } else if (iDotCount > 1) {  
            break LABELinner;  
        } else {  
            blsNum = true;  
        }  
    } else {  
        blsNum = false;  
        break LABELinner;  
    }  
}  
  
iDotCount = 0;  
  
dHRrest = Double.parseDouble(sInput);  
  
if (dHRrest < 35) {  
    System.out.print("Are you dead?\n");  
    break LABELinner;  
} else {  
    break LABELtestForNum;  
}
```

```
    }  
}
```

```
System.out.print("Please enter a valid heart rate that is more than 35 > ");  
sInput = input.nextLine();  
bIsNum = true;  
iDotCount = 0;  
}
```

```
System.out.print("Enter your age > ");  
sInput = input.nextLine();
```

```
LABELtestForNum: while (!bHasInput) {  
    LABELinner: while (bIsNum) {  
        for (int i = 0; i < sInput.length(); i++) {  
            Boolean bFlag = Character.isDigit(sInput.charAt(i));  
            if (bFlag) {  
                bIsNum = true;  
            } else if (sInput.charAt(i) == '.') {  
                iDotCount++;  
  
                if (i == 0) {  
                    break LABELinner;  
                } else if (iDotCount > 1) {  
                    break LABELinner;  
                } else {  
                    bIsNum = true;  
                }  
            }  
        }  
    }  
}
```

```
    }  
    } else {  
        blsNum = false;  
        break LABELinner;  
    }  
}
```

```
iDotCount = 0;
```

```
dAge = Double.parseDouble(sInput);
```

```
if (dAge < 4) {  
    System.out.print("Please enter a real age.\n");  
    break LABELinner;  
} else {  
    break LABELtestForNum;  
}  
}
```

```
System.out.print("Please enter a valid age that is more than 3 > ");  
sInput = input.nextLine();  
blsNum = true;  
iDotCount = 0;  
}
```

```
System.out.print("Enter your gender [m/f] > ");  
sInput = input.nextLine();
```

```
    LABELtestForChar: while (!bHasInput) {  
        LABELinner: while (bIsChar) {  
            if (sInput.length() > 1) {  
                bIsChar = false;  
                break LABELinner;  
            } else {  
                cGender = sInput.charAt(0);  
                bIsChar = true;  
  
                if (cGender == 'm' || cGender == 'M') {  
                    cGender = 'M';  
                    break LABELtestForChar;  
                } else if (cGender == 'f' || cGender == 'F') {  
                    cGender = 'F';  
                    break LABELtestForChar;  
                } else {  
                    break LABELinner;  
                }  
            }  
        }  
    }  
  
    System.out.print("Please enter a valid gender [m/f] > ");  
    sInput = input.nextLine();  
    bIsChar = true;  
}
```

```
dHRmax = 220 - dAge;
```

```
dVOMax = 15 * (dHRmax / dHRrest);
```

```
System.out.println("VO2Max > " + dVOMax); // test check print out
```

```
if (dAge < 10) {
```

```
    System.out.print("You're too young to take part in any sport.\n");
```

```
} else if (dAge > 30) {
```

```
    System.out.print("You're too old to take part in any sport.\n");
```

```
}
```

```
if (dAge >= 18 && dAge <= 30) {
```

```
    if (cGender == 'M' && (dVOMax >= 40 && dVOMax <= 60)) {
```

```
        iSportCount++;
```

```
        System.out.print("  Mens Basketball\n");
```

```
    } else if (cGender == 'F' && (dVOMax >= 43 && dVOMax <= 60)) {
```

```
        iSportCount++;
```

```
        System.out.print("  Womens Basketball\n");
```

```
    }
```

```
}
```

```
if (dAge >= 18 && dAge <= 26) {
```

```
    if (cGender == 'M' && (dVOMax >= 62 && dVOMax <= 74)) {
```

```
        iSportCount++;
```

```
        System.out.print("  Mens Cycling\n");
```

```
    } else if (cGender == 'F' && (dVOMax >= 47 && dVOMax <= 57)) {
```

```
        iSportCount++;
```

```

        System.out.print("  Womens Cycling\n");
    }

    if (cGender == 'M' && (dVOMax >= 55 && dVOMax <= 67)) {
        iSportCount++;
        System.out.print("  Mens Canoeing\n");
    } else if (cGender == 'F' && (dVOMax >= 47 && dVOMax <= 67)) {
        iSportCount++;
        System.out.print("  Womens Canoeing\n");
    }
}

if (dAge >= 18 && dAge <= 22) {
    if (cGender == 'M' && (dVOMax >= 52 && dVOMax <= 58)) {
        iSportCount++;
        System.out.print("  Mens Gymnastics\n");
    } else if (cGender == 'F' && (dVOMax >= 36 && dVOMax <= 50)) {
        iSportCount++;
        System.out.print("  Womens Gymnastics\n");
    }
}

if (dAge >= 10 && dAge <= 25) {
    if (cGender == 'M' && (dVOMax >= 50 && dVOMax <= 70)) {
        iSportCount++;
        System.out.print("  Mens Swimming\n");
    } else if (cGender == 'F' && (dVOMax >= 40 && dVOMax <= 60)) {

```

```

        iSportCount++;

        System.out.print("  Womens Swimming\n");
    }
}

if (iSportCount == 0) {
    System.out.print("You do not qualify for any type of sport.\n");
} else if (iSportCount == 1) {
    System.out.printf("You qualify for 1 sport.\n");
} else if (iSportCount > 0) {
    System.out.printf("You qualify for %d sports.\n", iSportCount);
}
}
}

//=====

```

E) Code documentation and explanation

The way my program works is by first taking the user's input as a string in the next line to check the whole input. This is to check for alphabets, symbols and spaces per character in the string variable. The reason for storing the input as a string is that, if an input contains any characters, it will crash the program if the variable type is a double or integer as those characters cannot be stored as those variable types. Hence, the input is a string to store the input and to check for these illegal characters with an alphanumeric checker.

The alphanumeric checker works the same as question 1 and 2 which is via a for loop (which is inside a while loop with a condition that the input is still a number after each character that is checked and is not an illegal character), where the maximum value of "i" is less than the input length. Each character from left to right is passed through a boolean to check if it is a digit. If the current character that is being checked is not a digit, it will break out from the loop and set the boolean for the input being a number to false. Then reprompts the user to reenter a new and valid digit.

However, numbers have decimal points and negative numbers, so we have to create a special check list for these cases. For negative numbers, we just need to check if the value of "i" in the for loop is 0. if not, if the character at i is '-' (as this means that the input is -> 123-123, where the '-' is not the first character, or -123 where the first character is a '-' but the second character is also a '-' and is therefore not a valid number), we break from the loop and reprompt for a new value. For decimal numbers, we first check if it's the first character at "i" is 0, if it is, we break and reprompt for a new value. If its not the first character, we add a dot counter to check how many dots the string contains. If it's the first one, the dot check will be 1 and its still a valid number. However, if the string contains a second dot, when the counter for the amount of dots is more than 1, it will break from the loop and reprompt the user for a new number.

I have tried my best to account for as many special cases to return as error casses, but there may be more that I cannot think of. For the most part, the alphanumeric checker works fine for what I need it to do.

Else, one all validity checks are done and the inputs are valid, for the heart rate and age, we just need to check if the values are real numbers (checking if the pulse input is too low or if the age input is too low, which again are bases on google). If their pulse is less than 35 or their age is less than 4, they will be reprompted to enter a new value for either.

For the gender input, we take it in as a string and check if the input length is more than 1, if it is, then its not [m/f]. If the length is 1, then we check if the cahracter is 'm' or 'M' or 'f' or 'F'. Then we convert the input to upper or lower case for easier checking later on -> if (cGender == 'm' || cGender == 'M') { cGender = 'm'; }

After all the validity checks are done, we then calculate the VO2Max -> ($dHR_{max} = 220 - dAge$) then ($dVO_{max} = 15 * (dHR_{max} / dHR_{rest})$). Then, based on the cGender, dAge, and dVO2max values, we check which sport the user is eligible to participate in. For each sport that they meet the requirement on, we add 1 to a sport count (iSportCount++). So once we check which sport the user is able to be a part of, we print each of the sports they can participate in and we print the total amount of sports they can participate in. If they do not meet any requirements, then we print that they will not be able to take part in any sport.