Question 1 **(10 marks)**

Write a Java class, called `Fitness` to maintain information about a person's name, age, weight and height. A name comprises a first name, a middle initial and a last name.

Write a constructor for this class that initialises all attribute variables. *No error checking of parameters* is required for this question. In addition to the constructor, your class should implement the following three methods:

- `firstNameIs` takes a first name as a parameter and returns true if it is the same as this individual's name.
- `bmi` calculates the body mass index for this individual. The body mass index is a measure of relative weight defined as the individual's body mass (weight) divided by the square of their height with units of kg/m^2.
- `bmiNormal` returns true if the individual's BMI is in the normal range and false otherwise. A normal BMI is defined as a value between 18.5 and 25.

Include brief comments where necessary, but full Javadoc comments are *not* required.

Answer Question 1 on the next page

Question 2 (10 marks)

2a) Write a method `countEvens` that takes an `ArrayList` of `String` objects as input and returns the number of even length strings contained in the input. For example, if the input is `[ "one", "peach", "pear", "plum" ]` then `countEvents(inp)` should return 2.

2b) Write a method, `mirror`, that doubles the size of a list of integers by appending a mirror image of the list. For example, given an array list containing `[ 1, 5, 2, 6 ]`, the method should return an array list containing `[ 1, 5, 2, 6, 6, 2, 5, 1 ]`. Use the `ArrayList` data structure to represent the input and output of this method.

Answer question 2a here

Question 3 (10 marks)

Write a method, `season`, that takes two integers as parameters representing a month and day and that returns a String indicating the season for that month and day. Assume that months are specified as an integer between 1 and 12 (1 for January, 2 for February, and so on) and that the day of the month is a number between 1 and 31. You should check that the inputs are within these bounds, but you are *not* required to check for invalid month, day combinations such as month=2 and day=30.

> For dates between 16 December and 15 March, return "Summer".
> For dates between 16 March and 15 June, return "Autumn".
> For dates between 16 June and 15 September, return "Winter".
> For dates between 16 September and 15 December, return "Spring".

3a) Complete the table on the next page by adding five (5) new test cases to check the correctness of an implementation of this method. The first line (in italics) is given as an example. The Rationale column should contain a brief comment for each test case to justify your choice.

3b) Write Java code to implement the `season` method.

Answer Question 3 on the next page

Answer Question 3a here

month	day	Expected output	Rationale
3	**15**	*"Summer"*	*boundary case*

Question 4 (10 marks)

Consider the following figure constructed using (only) the characters `+ = |` space and newline.

```
+===+===+
|   |   |
|   |   |
|   |   |
+===+===+
|   |   |
|   |   |
|   |   |
+===+===+
```

4a) Write a Java method called `window` that produces this figure as output. *Use nested for-loops* to print the repeated parts of the figure.

4b) Explain how you would modify your method so that it takes an input parameter for size (e.g. size=3 in the above example) and produces a figure that is larger and wider proportionally. The size parameter should always be greater than or equal to 3. You do *not* need to write the code for this more general method, just explain how you would do it.

Answer Question 4 over the page

Question 5 (10 marks)

The methods `foo1` and `foo2` are intended to calculate the number of positive factors of a positive integer. For example, the six factors of 12 are 1, 2, 3, 4, 6, and 12, so `foo1(12)` or `foo2(12)` should return 6.

Compare and contrast the methods `foo1` and `foo2`. Comment on the strengths and weakness of each implementation, using the criteria of correctness, design, efficiency, and readability. Use the line numbers to refer to parts of the code in your answer.

```
 1  private int foo1(int n) {
 2    for (int i = 0; i <= n; i++) {
 3      if (n % i = 0) {
 4        return i;
 5      }
 6    }
 7  }
 8
 9
10  public int foo2(int n) {
11    int count = 1;
12    for (int i = 1; i < n; i++) {
13      if (n % i == 0) {
14        count = count + 1;
15      }
16    }
17    return count;
18  }
```