

Question 1

```
public class Fitness
{
    private String firstName;
    private char middleInitial;
    private String lastName;
    private int age;           // years
    private double weight, height; // kg, m

    public Fitness(String fn, char m, String ln, int a, double w, double h)
    {
        firstName = fn;
        middleInitial = m;
        lastName = ln;
        age = a;
        weight = w;
        height = h;
    }

    public boolean firstNames(String s)
    {
        return firstName.equals(s);
    }

    public double bmi()
    {
        return weight / (height * height);
    }

    public boolean bmiNormal()
    {
        double b = bmi();
        return 18.5 <= b && b <= 25;
    }
}
```

Question 2

```
public static int countEvens(ArrayList<String> xs)
{
    int z = 0;
    for (String s : xs)
        if (s.length() % 2 == 0) z++;
    return z;
}
```

```
public static ArrayList<Integer> mirror(ArrayList<Integer> xs)
{
    for (int k = xs.size() - 1; k >= 0; k--)
        xs.add(xs.get(k));
    return xs;
}
```

Question 3

Month	Day	Expected	Rationale
6	15	Autumn	Lower boundary
6	16	Winter	Upper boundary
2	1	Summer	Start of month
10	31	Spring	End of month
13	13	?	Error

```
public static String season(int m, int d)
{
    if (m < 1 || m > 12 || d < 1 || d > 31) return "error";
    if (m == 1 || m == 2 || m == 12 && d >= 16 || m == 3 && d <= 15) return "Summer";
    if (m == 4 || m == 5 || m == 3 && d >= 16 || m == 6 && d <= 15) return "Autumn";
    if (m == 7 || m == 8 || m == 6 && d >= 16 || m == 9 && d <= 15) return "Winter";
    if (m == 10 || m == 11 || m == 9 && d >= 16 || m == 12 && d <= 15) return "Spring";
    return "confused";
}
```

Question 4

```
public static void window()
{
    line(3, "+", "=");
    for (int i = 1; i < 3; i++)
    {
        for (int j = 0; j < 3; j++) line(3, "|", " ");
        line(3, "+", "=");
    }
}

public static void line(int k, String x, String y)
{
    String s = x;
    for (int i = 1; i < k; i++)
    {
        for (int j = 0; j < k; j++) s += y;
        s += x;
    }
    System.out.println(s);
}
```

Give window a parameter

int n;

and replace all instances of 3 with n.

Question 5

Correctness:

foo1 has several issues:

- = instead of == on Line 3
- loop starts from 0 instead of 1 on Line 2, so divides by 0
- returns smallest factor of n on Line 4
- no return statement outside conditional

foo2 is correct.

Design:

Both methods use the obvious method of checking (more-or-less) all numbers in the range 1..n. This iteration could be reduced (see below).

Efficiency:

Both methods could be more efficient.

- there's no need to check if 1 is a factor
- they could stop at \sqrt{n} and add 2 for each factor (except the square root itself 😊)

Readability:

Both methods follow usual readability guidelines, e.g. {} for all statement-bodies.