Name:	Christopher Chan
Student Number:	20143087
Module:	Software Development
Class:	HDSDEV_SEP

Continuous Assessment

Q1H.

1. Identify and document the input, process and output for the application.

- Ask the user for their membership status and whether they are flying inside or outside of Europe. (**input**)
- Perform the calculation on how many reward points the user will receive and if the user is entitled to priority boarding or not. (**process**)
- Display the users reward points and priority boarding status. (**output**)

2. *Develop the instantiable class according to the question specification….*

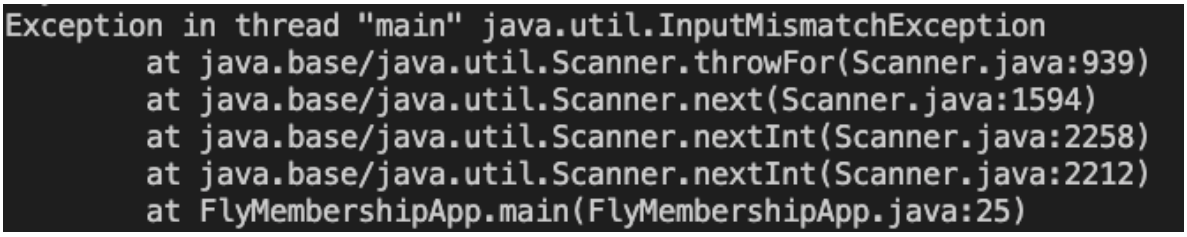
I developed the instantiable class with overloaded constructors to give the option to define the object variable FlyMembership with parameters (if the membership status and information about the user is flying in Europe is known). The application can also make a default object without this information and set the values at a later time using the setter methods setStatus(int) and setEuropeTravel(boolean).

3. *Develop the application class that uses the instantiable class previously developed.*

In the application class I used the default constructor and made a variable person1 of type FlyMembership. As the user inputs their membership status and information about their flight. The information is passed to the FlyMembership class and calculation is carried out by the calculateRewards() method. get-Points() and getPirorityBoarding() return information to the application.

4. *Document five examples of compilation errors and bugs and how you fixed them.*

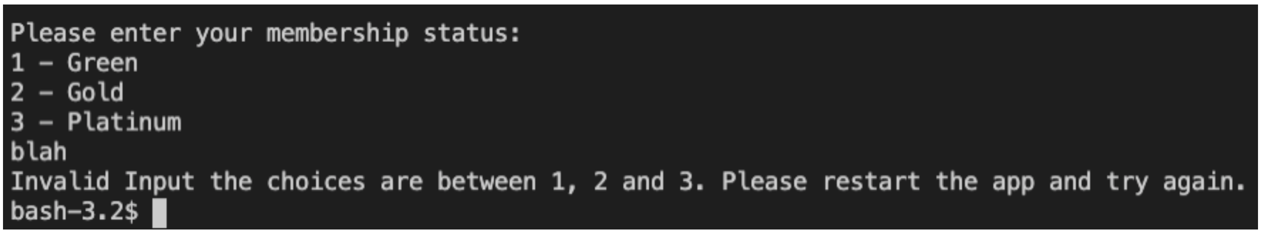
- In the FlyMembershipApp, in order to compensate for a non valid type input while the user is is selecting their membership status.



```
Exception in thread "main" java.util.InputMismatchException
	at java.base/java.util.Scanner.throwFor(Scanner.java:939)
	at java.base/java.util.Scanner.next(Scanner.java:1594)
	at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
	at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
	at FlyMembershipApp.main(FlyMembershipApp.java:25)
```

Invalid input type, Scanner obj throws an error.

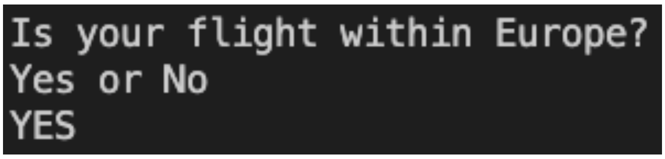
I put a try catch statement (ref1) to give the user feedback that an invalid input has been made and the steps to use the app correctly.



```
Please enter your membership status:
1 - Green
2 - Gold
3 - Platinum
blah
Invalid Input the choices are between 1, 2 and 3. Please restart the app and try again.
bash-3.2$ █
```

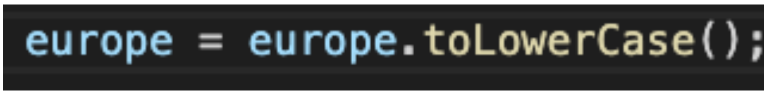
The app gives the user instructions on proper use of the app.

- When the application prompts the user “Is your flight within Europe?”, to compensate for the uppercase input .toLowerCase() was used.



```
Is your flight within Europe?
Yes or No
YES
```

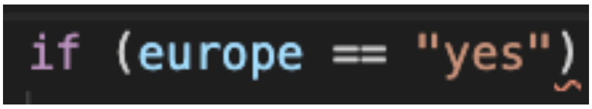
To compensate for the Uppercase input



```
europe = europe.toLowerCase();
```

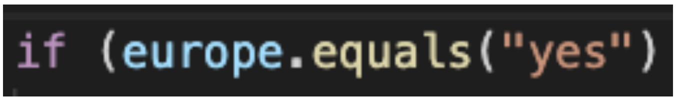
The inputted string is cast to lowercase

- When comparing strings in the if conditional statement, instead of using the comparison operator == the . equals(String) method was used to successfully compare two strings.



```
if (europe == "yes")
```

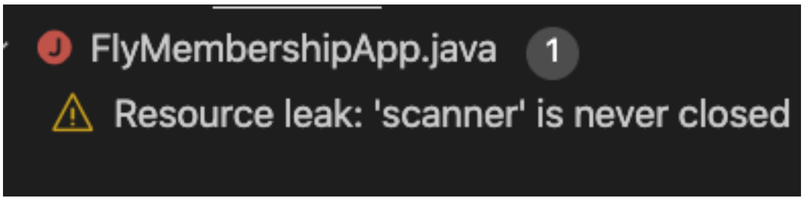
Incorrect comparison for Strings



```
if (europe.equals("yes")
```

Using the .equals(String) method

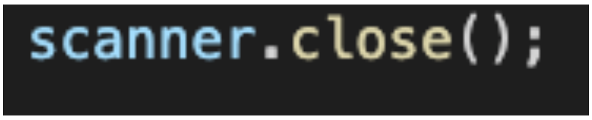
- Since the scanner object is used in the application it is good practice to close the object before the application terminates. Initially my IDE visual studio code complained but using the .close() on the scanner object fixed the warning.



FlyMembershipApp.java 1

⚠ Resource leak: 'scanner' is never closed

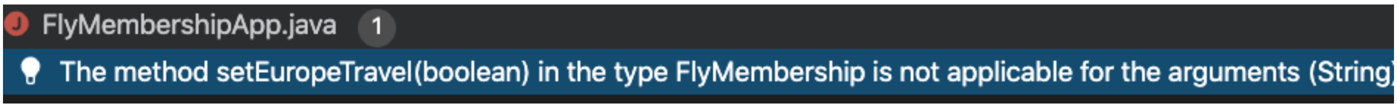
Warning shown by vs code



```
scanner.close();
```

Close the Scanner object

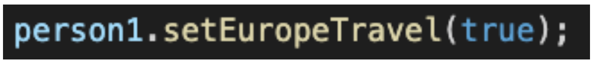
- Passing a string instead of the expected boolean was fixed.



FlyMembershipApp.java 1

The method setEuropeTravel(boolean) in the type FlyMembership is not applicable for the arguments (String

Incorrect parameter passed.



```
person1.setEuropeTravel(true);
```

Fixed boolean to true

Q2G

1. Identify and document the input, process and output for the application.

- Ask the user to input a sentence to encode (**input**)
- Encoding the inputted sentence with the characters ‘?’, ‘*’and ‘!’ (**process**)
- Display the encoded sentence to the user. (**output**)

2. *Develop the instantiable class according to the question specification….*

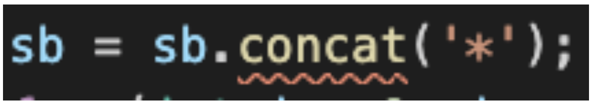
In the SentenceEncoder class an overloaded constructor and default instructor were defined. One setter and one getter method we made to encapsulate internal class data. One public method to do the calculation was also defined.

3. *Develop the application class that uses the instantiable class previously developed.*

In the application class a scanner object is used to read in the users input. A new object variable encode of type SentenceEncoder is made and the testSentence is passed to the object. A encodeSentence() method is called and the output is displayed to the user.

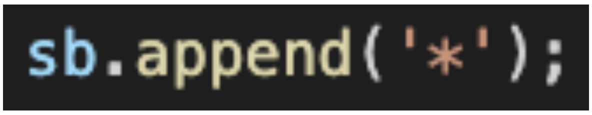
4. *Document five examples of compilation errors and bugs and how you fixed them.*

- In the encodeSentence() method initially trying to manipulate a String threw up errors because Strings are immutable objects. This was solved by using the mutual Object StringBuffer.



```
sb = sb.concat('*');
```

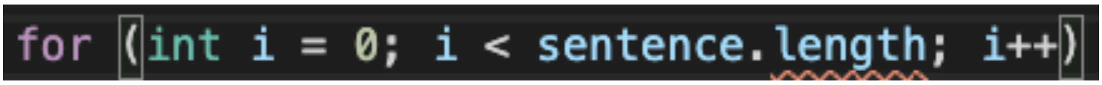
String sb cannot add the char (ref2)



```
sb.append('*');
```

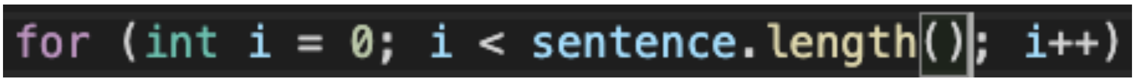
StringBuffer sb added the char

- In the encodeSentence() method using .length in the for loop threw up and error. .length is used to get the length of an array. To get the length of a String Object .length() was used.



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

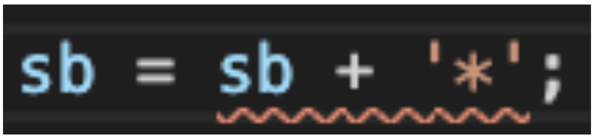
.length error



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

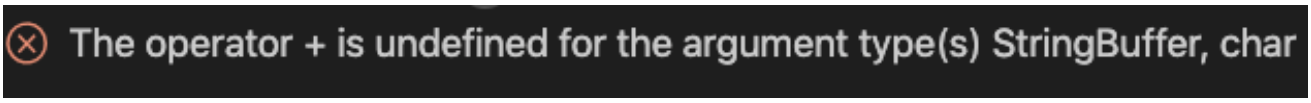
.length() fix

- When creating a StringBuffer Object, new characters cannot be concatenated using the + operator. New characters were appended instead to a StringBuffer object



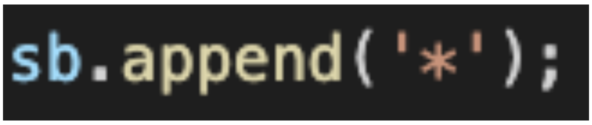
```
sb = sb + '*';
```

The + operator cannot be used.



```
The operator + is undefined for the argument type(s) StringBuffer, char
```

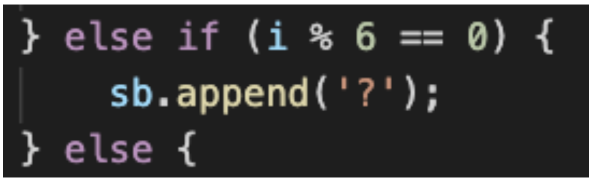
Compiler Error



```
sb.append('*');
```

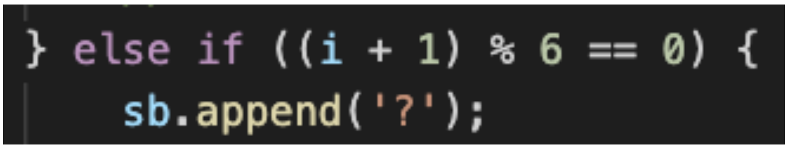
StringBuffer uses append to add (ref3)

- As per the specification each 6th character should be replace by a ‘?’ character, since I was iterating through the String and Strings start with 0 index. While appending my StringBuffer object I had to compensate with (i+1) when appending the ‘?’ character.



```
} else if (i % 6 == 0) {
    sb.append('?');
} else {
```

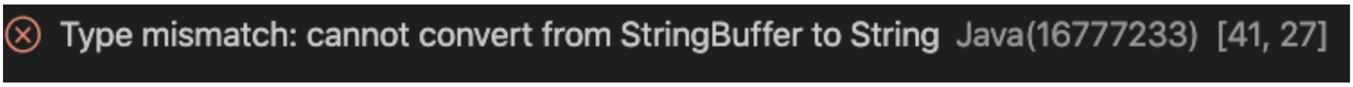
Logical error in if else



```
} else if ((i + 1) % 6 == 0) {
    sb.append('?');
```

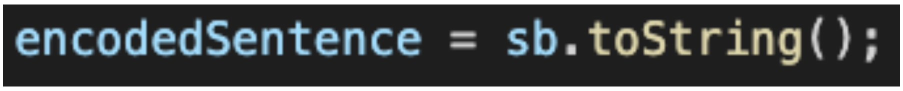
(i + 1) compensation

- Trying to assign the value of the StringBuffer object to a String threw an error. A toString() in the StringBuffer class to be used to cast to String.



```
ⓧ Type mismatch: cannot convert from StringBuffer to String  Java(16777233)  [41, 27]
```

Type mismatch



```
encodedSentence = sb.toString();
```

toString() method (ref4)

Appendix -

1. Try Catch Statement - https://www.w3schools.com/java/java_try_catch.asp
2. String Concat - https://www.w3schools.com/java/ref_string_concat.asp
3. StringBuffer append - https://www.geeksforgeeks.org/stringbuffer-append-method-in-java-with-examples/
4. StringBuffer to String - https://www.geeksforgeeks.org/stringbuffer-tostring-method-in-java-with-examples/

Source Code

```java
public class FlyMembership {

    // Our Private data Variables
    // Stores the points
    private int points;
    // Status variable
    private int status;
    // Checks if the customer is flying in or out of europe
    private boolean europe;
    // Variable to check if customer is allowed Priority Boarding
    private boolean priorityBoarding;

    // ------------Constructors ------------
    // Default constructor
    public FlyMembership() {

    }

    // Overload the constructor, takes the status in the parameters
    public FlyMembership(int status) {
        this.status = status;
    }
    // Overload the constructor, takes the status and europe
travel in the
    // parameters
    public FlyMembership(int status, boolean europe) {
        this.status = status;
        this.europe = europe;
    }

    // ------------Getters and setters ------------

    //method to pass the membership status from the user
    public void setStatus(int status) {
        this.status = status;
    }

    //method to pass if the user is travelling in europe
    public void setEuropeTravel(boolean europe) {
        this.europe = europe;
    }

    // get the points
    public int getPoints() {
        return points;
    }

    // get if pirority boarding is allowed
    public String getPirorityBoarding() {
        if (priorityBoarding) {
            return "Yes";
        }
        return "No";
    }

    // ------------- public methods-------------

    // This method does the processing
    public void calculateRewards() {
        // If Membership status is Green
        if (status == 1) {
            // if user travelled in europe
            if (europe == true) {
                points = points + 10;
            } else {
                points = points + 20;
            }
            // If Membership status is gold
        } else if (status == 2) {
            // if user travelled in europe
            if (europe == true) {
                points = points + 30;
            } else {
                points = points + 50;
                priorityBoarding = true;
            }
            // If Membership status is platiumn
        } else if (status == 3) {
            // if user travelled in europe
            if (europe == true) {
                points = points + 70;
            } else {
                points = points + 90;
            }
            priorityBoarding = true;
        }
    }

}
```

```
import java.util.Scanner;

public class FlyMembershipApp {
    public static void main(String[] args) {
        // Declare variables
        int status;
        String europe;

        // Lets declare an Object variable of Type FlyMembership
using the default
        // constructor
        FlyMembership person1 = new FlyMembership();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Welcome to Fly Club!");

        while (true) {
            // Lets catch bad inputs (not int)
            try {
                System.out.println("Please enter your membership
status: ");
                System.out.println("1 - Green");
                System.out.println("2 - Gold");
                System.out.println("3 - Platinum");

                // -----------the input -----------
                status = scanner.nextInt();
                if (status >= 1 && status <= 3) {

                    // Set the status variable in the Flymember-
ship Obj

                    person1.setStatus(status);
                    // lets pass the membership status value to
our FlyMembership Obj
                    System.out.println("Is your flight within Eu-
rope?");

                    System.out.println("Yes or No");

                    // -----------the input -----------
                    europe = scanner.next();
                    europe = europe.toLowerCase();

                    // validates input is "yes" or "y"
```

```java
                    if (europe.equals("yes") || europe.equals("y")) {
                        // ----------------The processing--------------
                        person1.setEuropeTravel(true);
                        person1.calculateRewards();
                        // validates input is "no" or "n"
                    } else if (europe.equals("no") || europe.equals("n")) {
                        // ----------------The processing--------------
                        person1.calculateRewards();
                    } else {
                        // error is input other than yes or no is given
                        System.out.println("Invalid yes or no input, Restart the app and try again.");
                    }
                    // -------------------Output-------------------
                    System.out.println("Reward Points: " + person1.getPoints());
                    System.out.println("Piroirty Boarding: " + person1.getPirorityBoarding());
                    break;
                }
            } catch (Exception e) {
                System.out.println(
                        "Invalid Input the choices are between 1, 2 and 3. Please restart the app and try again.");
                break;
            }
        }
        // Close the Scanner Obj, so no memory leaks
        scanner.close();
    }
}
```

```
public class SentenceEncoder {

    // Encapulated internal data
    private String sentence;
    private String encodedSentence;

    // Dont really use the defualt constructor in this app....
    public SentenceEncoder() {

    }

    public SentenceEncoder(String sentence) {
        this.sentence = sentence;
    }

    // ----------Setters and Getters-------------
    // if the deafult constructor is used we can pass a sentence
using this setter
    public void setSentence(String sentence) {
        this.sentence = sentence;
    }

    // return the encoded sentence
    public String getEncodedSentence() {
        return encodedSentence;
    }

    // ------------Methods of SentenceEncoder---------

    public void encodeSentence() {

        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < sentence.length(); i++) {
            // if there is a space ' ' add '*' to sb
            if (sentence.charAt(i) == ' ') {
                sb.append('*');
                // if there is a '.' add '!' to sb
            } else if (sentence.charAt(i) == '.') {
                sb.append('!');
                // Because string starts at 0 we use (i+1) to
find the 6th character in
                // intervals
                // mulitple of 6 add '?'
            } else if ((i + 1) % 6 == 0) {
                sb.append('?');
                // every other character add to the sb
            } else {
                sb.append(sentence.charAt(i));
            }
        }
        encodedSentence = sb.toString();
    }

}
```

```
import java.util.Scanner;

public class SentenceEncoderApp {
    public static void main(String[] args) {

        // Variable declaration
        SentenceEncoder encode;
        Scanner scanner = new Scanner(System.in);
        String testSentence;

        System.out.println("Input a sentence to encode:");
        // ---------------input--------------
        testSentence = scanner.nextLine();
        // -------------process--------------
        // pass the input testsentence to the SentenceEncoder
class
        encode = new SentenceEncoder(testSentence);
        // encode the sentence
        encode.encodeSentence();

        // -------------output --------------
        System.out.println(encode.getEncodedSentence());
        // Close scanner.
        scanner.close();
    }
}
```