

Question one

Here is the rubric

- The overall fun aspect of playing the game - the level of creativity, originality etc (3)
- The creative design - background images, fonts, colours, layout etc (3)
- The flow from one screen to the next (3)
- The code in the UI class (3)
- The code in the manager class - comments, layout, variable names, whitespace etc (5)
- The level of sophistication found in the code, with regard to the game itself, random number selection etc (3)
- The code that reads in the text file (3)
- Bonus marks will be awarded if the program writes the winners details to a new text file (3)

Total: **23 marks** (plus a possible 3 bonus marks)

Grade 11

Memo. Nov Prac  
2022



-- File generated with SQLiteStudio v3.3.3 on Thu Oct 27 17:08:35 2022  
--  
-- Text encoding used: Shift\_JIS  
--

PRAGMA foreign\_keys = off;  
BEGIN TRANSACTION;

-- Table: tblBirds

CREATE TABLE tblBirds (birdID INTEGER PRIMARY KEY AUTOINCREMENT, birdname VARCHAR(255) NOT NULL, height INTEGER, wingspan INTEGER, eggs INTEGER, broods INTEGER, incubation INTEGER, fledging INTEGER, nestBuilder CHAN);  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (1, 'Great Blue Heron', 52, 78, 5, 1, 28, 60, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (2, 'Mallard', 28, 3, 10, 1, 30, 52, 'F');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (3, 'Common Loon', 36, 54, 2, 1, 31, 80, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (4, 'Bald Eagle', 37, 84, 2, 1, 36, 90, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (5, 'Golden Eagle', 40, 90, 3, 1, 45, 80, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (6, 'Red Tailed Hawk', 25, 48, 3, 1, 35, 46, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (7, 'Osprey', 24, 72, 4, 1, 42, 58, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (8, 'Belted Kingfisher', 13, 23, 7, 1, 24, 24, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (9, 'Canadian Goose', 43, 72, 10, 1, 30, 55, 'F');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (10, 'Pied-billed Grebe', 13, 7, 7, 1, 24, 24, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (11, 'American Coot', 16, 29, 12, 1, 25, 52, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (12, 'Common Sea Gull', 18, 18, 3, 1, 28, 36, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (13, 'Ring-billed Gull', 19, 50, 4, 1, 21, 40, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (14, 'Double-crested Cormorant', 33, 54, 4, 1, 29, 42, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (15, 'Common Merganser', 27, 34, 11, 1, 33, 80, 'F');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (16, 'Turkey Vulture', 32, 72, 2, 1, 41, 88, 'N');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (17, 'American Crow', 18, 40, 6, 1, 18, 35, 'F');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (18, 'Green Heron', 22, 27, 4, 2, 25, 36, 'B');  
INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (19,

```

30 'Mute Swan', 60, 95, 8, 1, 40, 150, 'B');
31 INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (20,
32 'Brown Pelican', 54, 90, 4, 1, 30, 77, 'F');
33 INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (21,
34 'Great Egret', 38, 67, 3, 1, 26, 49, 'B');
35 INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (22,
36 'Anhinga', 35, 42, 4, 1, 30, 42, 'B');
37 INSERT INTO tblBirds (birdID, birdname, height, wingspan, eggs, broods, incubation, fledging, nestBuilder) VALUES (23,
38 'Black Skimmer', 20, 15, 5, 1, 25, 30, 'B');

```

```

39 -- Table: tblFood
40 CREATE TABLE tblFood (foodID INTEGER PRIMARY KEY AUTOINCREMENT, foodName VARCHAR);

```

```

41 INSERT INTO tblFood (foodID, foodName) VALUES (1, 'Seeds');
42 INSERT INTO tblFood (foodID, foodName) VALUES (2, 'Birds');
43 INSERT INTO tblFood (foodID, foodName) VALUES (3, 'Fruit');
44 INSERT INTO tblFood (foodID, foodName) VALUES (4, 'Frogs');
45 INSERT INTO tblFood (foodID, foodName) VALUES (5, 'Fish');
46 INSERT INTO tblFood (foodID, foodName) VALUES (6, 'Berries');
47 INSERT INTO tblFood (foodID, foodName) VALUES (7, 'Aquatic Plants');
48 INSERT INTO tblFood (foodID, foodName) VALUES (8, 'Aquatic insects');
49 INSERT INTO tblFood (foodID, foodName) VALUES (9, 'Worms');
50 INSERT INTO tblFood (foodID, foodName) VALUES (10, 'Nuts');
51 INSERT INTO tblFood (foodID, foodName) VALUES (11, 'Rodents');
52 INSERT INTO tblFood (foodID, foodName) VALUES (12, 'Snakes');
53 INSERT INTO tblFood (foodID, foodName) VALUES (13, 'Small mammals');
54 INSERT INTO tblFood (foodID, foodName) VALUES (14, 'Nectar');
55 INSERT INTO tblFood (foodID, foodName) VALUES (15, 'Pollen');
56 INSERT INTO tblFood (foodID, foodName) VALUES (16, 'Carriion');
57 INSERT INTO tblFood (foodID, foodName) VALUES (17, 'Moths');
58 INSERT INTO tblFood (foodID, foodName) VALUES (18, 'Ducks');
59 INSERT INTO tblFood (foodID, foodName) VALUES (19, 'Insects');
60 INSERT INTO tblFood (foodID, foodName) VALUES (20, 'Plants');
61 INSERT INTO tblFood (foodID, foodName) VALUES (21, 'Corn');
62 INSERT INTO tblFood (foodID, foodName) VALUES (22, 'Crayfish');
63 INSERT INTO tblFood (foodID, foodName) VALUES (23, 'Crustaceans');
64 INSERT INTO tblFood (foodID, foodName) VALUES (24, 'Repiles');
65 INSERT INTO tblFood (foodID, foodName) VALUES (25, 'Deer');

```

```

66 -- Table: tblFoodBird
67 CREATE TABLE tblFoodBird (birdID REFERENCES tblBirds (birdID), foodID REFERENCES tblFood (foodID));

```

```

68 INSERT INTO tblFoodBird (birdID, foodID) VALUES (1, '5');
69 INSERT INTO tblFoodBird (birdID, foodID) VALUES (1, '4');
70 INSERT INTO tblFoodBird (birdID, foodID) VALUES (1, '19');
71 INSERT INTO tblFoodBird (birdID, foodID) VALUES (1, '12');
72 INSERT INTO tblFoodBird (birdID, foodID) VALUES (2, '1');
73 INSERT INTO tblFoodBird (birdID, foodID) VALUES (2, '20');
74 INSERT INTO tblFoodBird (birdID, foodID) VALUES (2, '8');
75 INSERT INTO tblFoodBird (birdID, foodID) VALUES (2, '21');

```

73 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('3', '5');  
74 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('3', '8');  
75 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('4', '5');  
76 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('4', '16');  
77 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('4', '18');  
78 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('5', '13');  
79 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('5', '5');  
80 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('5', '24');  
81 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('5', '2');  
82 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('5', '19');  
83 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('5', '25');  
84 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('6', '2');  
85 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('6', '12');  
86 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('6', '11');  
87 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('6', '19');  
88 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('7', '5');  
89 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('8', '5');  
90 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('9', '7');  
91 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('9', '19');  
92 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('9', '1');  
93 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('10', '8');  
94 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('11', '8');  
95 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('11', '7');  
96 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('12', '5');  
97 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('12', '23');  
98 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('12', '16');  
99 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('13', '19');  
100 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('13', '5');  
101 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('13', '16');  
102 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('14', '5');  
103 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('14', '8');  
104 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('15', '5');  
105 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('15', '8');  
106 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('16', '16');  
107 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('17', '3');  
108 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('17', '8');  
109 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('17', '13');  
110 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('17', '5');  
111 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('17', '16');  
112 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('17', '1');  
113 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('18', '5');  
114 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('18', '19');  
115 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('18', '7');  
116 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('19', '8');  
117 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('19', '20');  
118 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('50', '5');  
119 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('20', '23');  
120 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('21', '5');

```
121 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('21', '8');
122 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('21', '4');
123 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('21', '22');
124 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('22', '5');
125 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('22', '8');
126 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('23', '5');
127 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('23', '23');
128 INSERT INTO tblFoodBird (birdID, foodID) VALUES ('23', '8');
129
130 COMMIT TRANSACTION;
131 PRAGMA foreign_keys = on;
132
```

```
1 /*
2  * Question 2.1 to 2.3
3  */
4
5 package questiontwo;
6
7 import java.io.File;
8 import java.io.FileNotFoundException;
9 import java.time.LocalDate;
10 import java.util.Scanner;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13 import javax.swing.JOptionPane;
14
15 public class QuestionTwo {
16
17     private static String theFile = null;
18     private static int size = 0;
19     private static String[] stringArr = new String[20];
20     private static LocalDate[] dateArr = new LocalDate[20];
21     private static LocalDate theDate = null;
22
23     public static void main (String[] args){
24
25         theFile = JOptionPane.showInputDialog("Enter the name of the text
26         file.");
27         readFile();
28         printStringArray();
29         printDateArray();
30
31         System.out.println("Main");
32     }
33
34     public static void readFile(){
35
36         String line = null;
37         int year = 0, month = 0, day = 0;
38
39         try {
40             Scanner scFile = new Scanner(new File(theFile));
```

```
41         while(scFile.hasNext()){ ✓
42
43             line = scFile.nextLine(); ✓
44             stringArr[size] = line; ✓
45
46             year = Integer.parseInt(line.substring(0,4)); ✓
47             month = Integer.parseInt(line.substring(5,7)); ✓
48             day = Integer.parseInt(line.substring(8)); ✓
49
50             theDate = LocalDate.of(year, month, day); ✓
51             dateArr[size] = theDate;
52             System.out.println("Dste is " + theDate);
53             size++;
54
55         } // end while loop
56
57     } catch (FileNotFoundException ex) {
58         Logger.getLogger(QuestionTwo.class.getName()).log(Level.SEVERE,
59 null, ex);
60         System.out.println("File not found"); ✓
61     }
62 } // end readfile
63
64 public static void printStringArray(){ ✓
65     for (int i = 0; i < size; i++) { ✓
66
67         System.out.println(stringArr[i]); ✓
68     } // end for
69
70 } // end printArray
71
72
73 public static void printDateArray(){
74
75     for (int i = 0; i < size; i++) {
76
77         System.out.println("Date array " + dateArr[i]); ✓
78     } // end for
79
80 } // end printArray
```

1 /\*  
2 \* Question four. Two class program with a UI and manager class  
3 \* Reads in a text file with delimiters  
4 \*/  
5 package questionthree;  
6  
7 public class NationalityUI {  
8  
9 public static void main(String[] args) {  
10 NationalityManager nm = new NationalityManager();  
11 double amountRaised = 0.0;  
12  
13 nm.readFile();  
14 nm.displayZA();  
15 nm.displayBO();  
16 amountRaised = nm.getAmount();  
17  
18 System.out.println("=====");  
19 System.out.println("Total Amount - ZA and BO - R" + amountRaised);  
20 }  
21 }

Q4



```
1 /*
2  * Question four. Two class program with a UI and manager class
3  * Reads in a text file with delimiters
4  * No constructor method - uses the default
5  */
6 package questionthree;
7
8 import java.io.File;
9 import java.io.FileNotFoundException; ✓
10 import java.util.Scanner;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13
14 public class NationalityManager {
15
16     private String[] countryZA = new String[50]; ✓
17     private String[] countryBO = new String[50]; ✓
18     private static int counterZA = 0, counterBO = 0; } ✓
19     private static double totalAmount = 0.0;
20
21     private String passport = null, country = null, name = null;
22     private boolean tax = false;
23     private double amount = 0.0;
24
25     public void readFile() { ✓
26         // The constructor can be used instead of a regular method
27
28         try {
29             Scanner scFile = new Scanner(new File("nationality.txt")); ✓
30
31             while(scFile.hasNext()) {
32                 String line = scFile.nextLine(); ✓
33                 Scanner scLine = new Scanner(line).useDelimiter("#"); ✓
34                 passport = scLine.next();
35                 int dash = passport.indexOf("-"); ✓
36                 country = passport.substring(dash + 1, dash + 3); ✓
37
38                 if(country.equals("ZA")) { ✓
39                     name = scLine.next();
40                     name = name.toUpperCase(); ✓
41                     tax = scLine.nextBoolean();
```

```
42         amount = scLine.nextDouble();
43         totalAmount = totalAmount + amount; ✓
44         countryZA[counterZA] = passport + "\t" + "\t" + name +
"\t" + tax + "\t" + amount; ✓
45         counterZA++; ✓
46     }
47
48     if(country.equals("BO")) { ✓
49         name = scLine.next();
50         name = name.toUpperCase();
51         amount = scLine.nextDouble();
52         totalAmount = totalAmount + amount;
53         countryBO[counterBO] = passport + "\t" + name + "\t" +
amount;
54         counterBO++; ✓
55     }
56
57     } // end while
58
59     } catch (FileNotFoundException ex) {
60
61         Logger.getLogger(NationalityManager.class.getName()).log(Level.SEVERE, null,
ex);
62         System.out.println("Error with file"); ✓
63     }
64     System.out.println("readfile");
65     } // end readFile
66
67     public void displayZA() {
68
69
70         System.out.println("Country ZA");
71
72         System.out.println("=====");
73         for(int i = 0; i < counterZA; i++) { ✓
74             System.out.println(countryZA[i]); ✓
```

```
75         } // end for
76
77         System.out.println();
78     } // end displayZA
79
80     public void displayBO() {
81
82         System.out.println("=====");
83         System.out.println("Country BO");
84         System.out.println("=====");
85         for(int i = 0; i < counterBO; i++) {
86             System.out.println(countryBO[i]);
87         }
88
89         System.out.println();
90     } // end displayBO
91
92     public double getAmount()
93
94         return totalAmount;
95
96     } // end getAmount
97
98 }
```



1 /\*  
2 \* Question 2.1 to 2.3  
3 \*/  
4  
5 package questiontwo;  
6  
7 import java.io.File;  
8 import java.io.FileNotFoundException;  
9 import java.time.LocalDate;  
10 import java.util.Scanner;  
11 import java.util.logging.Level;  
12 import java.util.logging.Logger;  
13 import javax.swing.JOptionPane;  
14  
15 public class QuestionTwo {  
16  
17 private static String theFile = null;  
18 private static int size = 0;  
19 private static int size2 = 0;  
20 private static String[] stringArr = new String[20];  
21 private static LocalDate[] dateArr = new LocalDate[20];  
22 private static LocalDate[] dateArr2 = new LocalDate[20];  
23 private static LocalDate theDate = null;  
24  
25 public static void main (String[] args){  
26  
27 theFile = JOptionPane.showInputDialog("Enter the name of the text  
file.");  
28 readFile();  
29 printStringArray();  
30 printDateArray();  
31 oldest();  
32  
33 readFile2();  
34 printDateArr2();  
35  
36 System.out.println("Main");  
37 }  
38  
39 public static void readFile(){  
40

Q5

```
41     String line = null;
42     int year = 0, month = 0, day = 0;
43
44     try {
45         Scanner scFile = new Scanner(new File("theFile.txt"));
46
47         while(scFile.hasNext()){
48
49             line = scFile.nextLine();
50             stringArr[size] = line;
51
52             year = Integer.parseInt(line.substring(0,4));
53             month = Integer.parseInt(line.substring(5,7));
54             day = Integer.parseInt(line.substring(8));
55
56             theDate = LocalDate.of(year, month, day);
57             dateArr[size] = theDate;
58
59             size++;
60
61             } // end while loop
62
63         } catch (FileNotFoundException ex) {
64
65             Logger.getLogger(QuestionTwo.class.getName()).log(Level.SEVERE, null, ex);
66             System.out.println("File not found");
67         }
68     } // end readfile
69
70     public static void printStringArray(){
71
72         for (int i = 0; i < size; i++) {
73
74             System.out.println("String array " + stringArr[i]);
75         } // end for
76     } // end printArray
77
78
79     public static void printDateArray(){
80
```

```
81      System.out.println("");
82      for (int i = 0; i < size; i++) {
83
84          System.out.println("Date array 1: " + dateArr[i]);
85      } // end for
86
87  } // end printArray
88
89  public static void oldest(){
90
91      System.out.println("");
92      LocalDate temp = LocalDate.now();
93
94      for(int i = 0; i < size; i++){
95          if(dateArr[i].isBefore(temp))
96              temp = dateArr[i];
97      }
98
99      System.out.println("The oldest date is " + temp);
100
101  } // end oldest
102
103  //=====
104
105  public static void readFile2(){
106
107      String line = null;
108      int year = 0, month = 0, day = 0;
109
110      try {
111          Scanner scFile = new Scanner(new File("newfile2.txt"));
112
113          while(scFile.hasNext()){
114
115              line = scFile.nextLine();
116              year = Integer.parseInt(line.substring(0,4));
117              String monthDay = line.substring(5);
118              int newDash = monthDay.indexOf("-");
119              month = Integer.parseInt(monthDay.substring(0,newDash));
120              day = Integer.parseInt(monthDay.substring(newDash + 1));
```

```
122
123         theDate = LocalDate.of(year, month, day);
124         dateArr2[size2] = theDate;
125
126         size2++;
127
128     } // end while loop
129
130     } catch (FileNotFoundException ex) {
131
132         Logger.getLogger(QuestionTwo.class.getName()).log(Level.SEVERE, null, ex);
133         System.out.println("File not found");
134     }
135 } // end readfile2
136
137 public static void printDateArr2(){
138     System.out.println("");
139     for (int i = 0; i < size2; i++) {
140
141         System.out.println("Date array 2: " + dateArr2[i]);
142     } // end for
143
144 } // end printArray2
145 }
```