```java
// Name: Leon Singleton
// User Name:aca16ls

import sheffield.*;
public class Assignment2 {
    public static void main(String[] args) {

//The constant values used throughout the program are stored
        final int POND_ROWS = 90;
        final int POND_COLUMNS = 360;
        final int DUCK_ROWS = 52;
        final int DUCK_COLUMNS = 130;
        final int POND_SCALE = 3;
        final int DUCK_SCALE = 2;

//creates an instance of the grphics window
        EasyGraphics g = new EasyGraphics(POND_COLUMNS*POND_SCALE,POND_ROWS*POND_SCALE);
//creates an instance so that the duck file can be read
        EasyReader fileInput = new EasyReader("duck.txt");
//two seperate character arrays store the conents of the pond and duck images
        char[][] duckArray = new char[DUCK_ROWS][DUCK_COLUMNS];
        char[][] pondArray = new char[POND_ROWS][POND_COLUMNS];


//loops through the first 6760 characters of the file
//each character is stored in the duck array with 52 rows and 130 columns
      for (int j=0 ; j < DUCK_ROWS; j++) {
            for (int i=0 ; i< DUCK_COLUMNS; i++) {
                char letter= fileInput.readChar();
                duckArray[j][i] = letter;
            }
      }

//loops through the next 32400 characters of the file
//each character is stored in the pond array with 90 rows and 360 columns
      for (int j=0 ; j < POND_ROWS; j++) {
```

```
37              for (int i=0 ; i< POND_COLUMNS; i++) {
38                  char letter= fileInput.readChar();
39                  pondArray[j][i] = letter;
40              }
41      }
42
43 //loops through the rows and columns of the pond array
44 //Each character in the pond array is checked to see what colour it represents
45 //depending on which character is verified a colour is set
46 //then using that colour a rectangle of  width 3 by height 3 is created
47 //the x and y coordiates are scaled by a multiple of 3
48 //the y coordiante is set by counting down from the highest row position
49          for (int j=0 ; j < POND_ROWS; j++) {
50              for (int i=0 ; i< POND_COLUMNS; i++) {
51                  if (pondArray[j][i]=='s') {
52                      g.setColor (135,206,235);
53                  }
54                  else if (pondArray[j][i]=='w') {
55                      g.setColor (0,102,102);
56                  }
57                  else if (pondArray[j][i]=='g') {
58                      g.setColor (153,255,51);
59                  }
60                  else if (pondArray[j][i]=='b') {
61                      g.setColor (102,51,0);
62                  }
63                  g.fillRectangle((i)*POND_SCALE,(POND_ROWS-j-1)*POND_SCALE,POND_SCALE,POND_SCALE);
64              }
65          }
66
67 //loops through the first 26 rows and columns of the duck array
68 //only the first 26 rows are checked so that a half duck is created
69 //Each character in the duck array is checked to see what colour it represents
70 //when an x is detected a 2x2 yellow rectangle is drawn
71      for (int j = 0; j < DUCK_ROWS/2; j++) {
72            for (int i =0; i < DUCK_COLUMNS; i++) {
```

```
73                  if (duckArray[j][i]=='x') {
74                      g.setColor (255,255,0);
75                      g.fillRectangle((i)*DUCK_SCALE,(DUCK_ROWS-j)*DUCK_SCALE,DUCK_SCALE,DUCK_SCALE);
76                  }
77              }
78          }
79
80  //loops through the rows and columns of the duck array 3 times to create 3 ducks
81  //a couunter is creater that increments after each duck is drawn, so that there
82  //is equal spacing between each duck
83          int increaseX =0;
84          for (int p = 0; p < 3; p++) {
85          increaseX = increaseX +125;
86              for (int j = 0; j < DUCK_ROWS/2; j++) {
87                  for (int i =0; i < DUCK_COLUMNS; i++) {
88                      if (duckArray[j][i]=='x') {
89                          g.setColor (255,255,0);
90                          g.plot((i+increaseX+100),(80-j));
91                      }
92                  }
93              }
94          }
95
96      }
97  }
```