Assignment #4

Name: Jenet Baribeau

Course: CECS 220-01-4168

Date: 11/2/2016

1. PP 8.1

   While the number isn't 51, enter in user data. Create an array with a limit of 50. Scan the input.

readInts.java

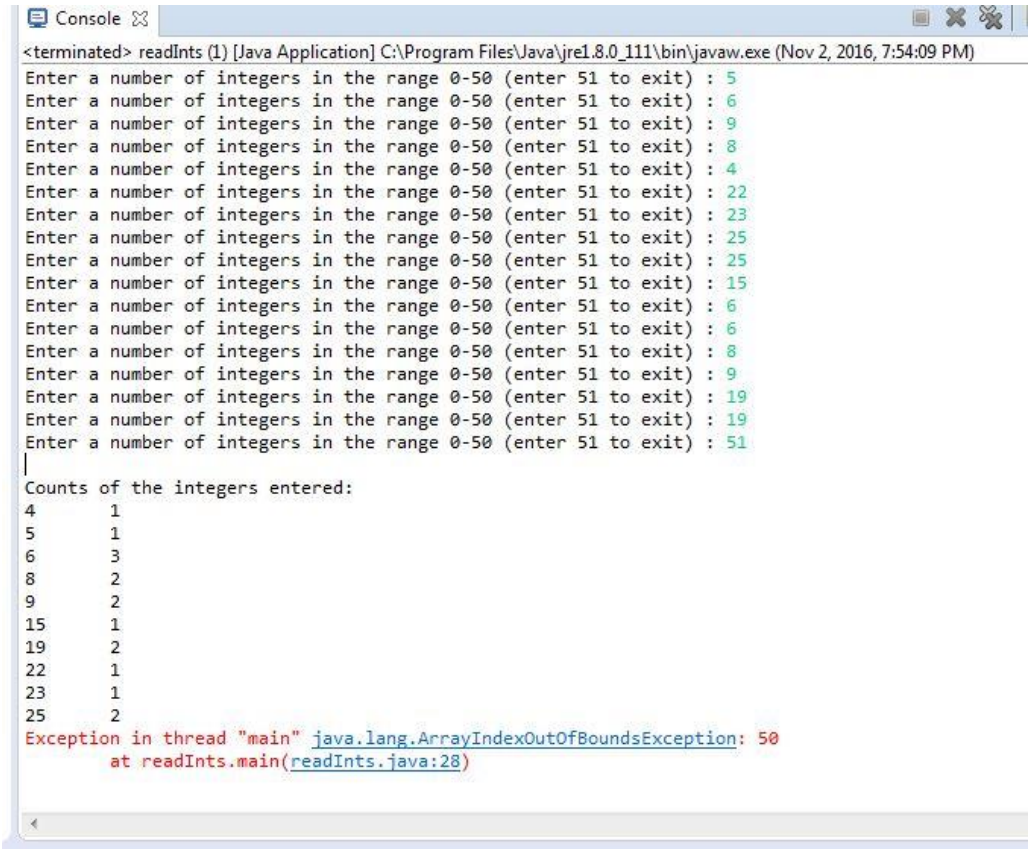
Assignment_4 ▸ src ▸ (default package) ▸ readInts ▸ main(String[]) : void

```
1  import java.util.Scanner;
2
3  public class readInts {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7          final int LIMIT = 50;
8          int input = 0;
9          int[] integers = new int[LIMIT];
10         Scanner scan = new Scanner (System.in);
11
12            while(input != 51)
13            {
14                 System.out.print("Enter an arbitrary number of integers in the range 0-50 (enter 51 to exit) : ");
15                 input = scan.nextInt();
16                 if(input == 51) break;
17                     if(input < 0 || input > 50){
18                         System.out.println("Number out of range.");
19                         }
20
```

   Continue until there is an integer enter over 51. Once entered for every integer less than 50, count and print them out.

```
                    }

                integers[input]++;
                   continue;
                    }

        System.out.println("\nCounts of the integers"
                + " entered: ");
                for (int i=0; i<=50; i++)
                if (integers[i]>0)
                System.out.println(i + "\t" + integers[i]);
            }
```

Output. I know I have an error but didn't use a try catch.



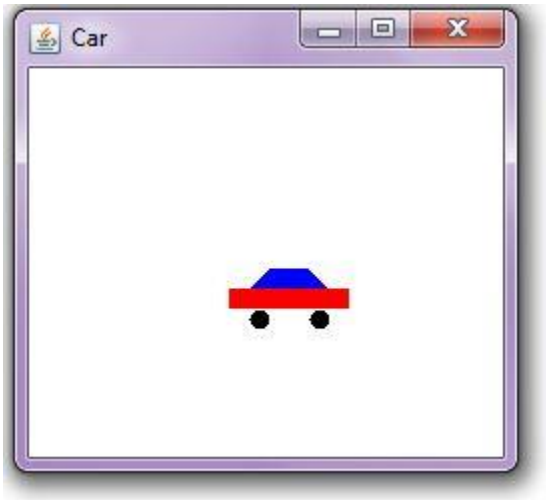
2. PP 8.13

```java
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;


    @SuppressWarnings("serial")
    public class carPanel extends JPanel
    {

      public void paint(Graphics g)
      {
    // setting background and foreground colors
         g.setColor(Color.white);
         g.fillRect(0, 0, getWidth(), getHeight());
         g.setColor(Color.red);

    // drawing the car body
         g.fillRect(100,110, 60, 10);

    // drawing the wheels
         g.setColor(Color.black);
         g.fillOval(110, 120, 10, 10);     // left wheel
         g.fillOval(140, 120, 10, 10);     // right wheel

         int x[] = {110, 120, 140, 150};   // coordinate arrays for the
         int y[] = {110, 100, 100, 110};   //   car cabin

         g.setColor(Color.blue);
         g.fillPolygon(x, y, 4);
      }


    }
```

Assignment_4 ▸ src ▸ (default package) ▸ car ▸

```java
    import javax.swing.JFrame;

    public class car
    {
        public static void main(String[] args)
        {
            JFrame frame = new JFrame("Car");
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            carPanel panel = new carPanel();
            frame.getContentPane().add(panel);
            frame.pack();
            frame.setVisible(true);
        }
    }
```

Finished Product:



3. PP9.2

I had a few issues with figuring out the %s and printf . There were several “Main” errors until I found the right combination.

Abstract Hospital Employee

```
1  package hospital;
2
3  public class HospitalEmployee {
4
5      protected String job;
6      protected String hospital;
7
8      public HospitalEmployee(String job, String hospital)
9      {
10         this.job = job;
11         this.hospital = hospital;
12
13     }
14
15     public void PerformJob(){
16
17     }
18
19 }
20
```

There are 6 extensions of the employee. This first one made the remainder very easy.

Administrator

```
package hospital;

public class Administrator  extends HospitalEmployee
{

    private final static String jobTitle = "Administrator";

    //Constructor to pass hospital of employee
    //requires super
    public Administrator(String hospital)
    {
        super(jobTitle, hospital);

    }

    //print the message
    public void PaperWork()
    {
        String message = String.format(
                "A %s does paperwork at %s.", job, hospital);
        System.out.println(message);
    }
    @Override
    public void PerformJob()
    {
        PaperWork();
    }


}
```

Hospital Driver

```
package hospital;

public class HospitalEmployees {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Creating an Administrator...");
        Administrator admin = new Administrator("Aurora BayCenter Hospital");
        System.out.println("Creating a Doctor...");
        Doctor doctor = new Doctor("University Hospital");
        System.out.println("Creating a Janitor employee...");
        Janitor janitor = new Janitor("St. Jude Hospital");
        System.out.println("Creating a Nurse employee...");
        Nurse nurse = new Nurse("Georgetown Hosptial");
        System.out.println("Creating a Receptionist employee...");
        Receptionist receptionist = new Receptionist("Hamptons Heritage");
        System.out.println("Creating a Surgeon employee...");
        Surgeon surgeon = new Surgeon("Baptist East Hospital");
        System.out.println();
        System.out.printf("Performing Job...");
        admin.PerformJob();
        System.out.printf("Performing Job...");
        doctor.PerformJob();
        System.out.printf("Performing Job...");
        janitor.PerformJob();
        System.out.printf("Performing Job...");
        nurse.PerformJob();
        System.out.printf("Performing Job...");
        receptionist.PerformJob();
        System.out.printf("Performing Job...");
        surgeon.PerformJob();

        System.out.printf("Performing Job...");
        doctor.PerformJob();
        System.out.printf("Performing Job...");
        janitor.PerformJob();
        System.out.printf("Performing Job...");
        nurse.PerformJob();
        System.out.printf("Performing Job...");
        receptionist.PerformJob();
        System.out.printf("Performing Job...");
        surgeon.PerformJob();

        System.out.println();
        System.out.println("The End.");
    }

}
```

Doctor

```java
package hospital;

public class Doctor extends HospitalEmployee {

    //static job title
    private final static String jobTitle = "Doctor";

    //constructor inherits employee
    public Doctor(String hospital)
    {
        super(jobTitle, hospital );
    }

    //Doctor's diagnose - print message
    public void Diagnose()
    {
        String message = String.format("A %s diagnoses patients at %s.",  job, hospital);
        System.out.println(message);


    }

    @Override
    public void PerformJob()
    {
        Diagnose();
    }
}

```

Surgeon

```java
package hospital;

public class Surgeon extends HospitalEmployee {

    // Static Job Title
    private final static String jobTitle = "Surgeon";


    // Constructor requires hospital of employee
    // passes to super constructor
    public Surgeon(String hospital) {
        super(jobTitle, hospital);
    }

    //prints surgery message
    public void Surgery() {
        String message = String.format(
                "A %s performs surgery at %s.", job, hospital);
        System.out.println(message);
    }

    @Override
    public void PerformJob() {
        Surgery();
    }
}
```

Nurse

```java
package hospital;

public class Nurse extends HospitalEmployee {

    // Static Job Title
    private final static String jobTitle = "Nurse";

    // Constructor requires hospital of employee
    // passes to super constructor
    public Nurse(String hospital) {
        super(jobTitle, hospital);
    }

    //prints take blood message
    public void TakeBlood() {
        String message = String.format(
                "A %s takes blood at %s.", job, hospital);
        System.out.println(message);
    }

    @Override
    public void PerformJob() {
        TakeBlood();
    }

}
```

Receptionist

```java
package hospital;

public class Receptionist extends HospitalEmployee {

    // Static Job Title
    private final static String jobTitle = "Receptionist";

    // Constructor requires hospital of employee
    // passes to super constructor
    public Receptionist(String hospital) {
        super(jobTitle, hospital);
    }

    //prints answer phone message
    public void AnswerPhones() {
        String message = String.format(
                "A %s answers phones at %s.", job, hospital);
        System.out.println(message);
    }

    @Override
    public void PerformJob() {
        AnswerPhones();
    }
}
```

Janitor

```java
package hospital;

public class Janitor extends HospitalEmployee {

    // Static Job Title
    private final static String jobTitle = "Janitor";

    // Constructor requires hospital of employee
    // passes to super constructor
    public Janitor(String hospital) {
        super(jobTitle, hospital);
    }

    //prints clean message
    public void Clean() {
        String message = String.format(
                "A %s cleans at %s.", job, hospital);
        System.out.println(message);
    }

    @Override
    public void PerformJob() {
        Clean();
    }
}
```

4. PP9.1

   The car moving was more challenging and far more complicated to make it move.

```
import java.awt.*;
import java.util.*;
import java.applet.*;



public class movingCar extends Applet implements Runnable
{
      Thread t;
      //4 variables used to vary the car's positions.
      int x1=0,x2=380,y1=50,y2=250;
      public void start()
      {
       if(t==null)
       {
        t=new Thread(this,"New Thread");//New side Thread created on start of applet.
        t.start();
       }
      }
      public void stop()
      {
       if(t!=null)
       {
        t=null;//On stop of applet the created thread is destroyed.
       }
      }
      //Implementation of method run() of Runnable interface.
      public void run()
      {
       Thread t1=Thread.currentThread();
       while(t==t1)
```