COP2800C Java 1 Module 11 Ungraded Practice Exercise

In this exercise we will practice using Dates and Times. You will be modifying an application for which I will provide the initial source files.

This application simulates a SpaceX™ Starship launch system and provides an Astronaut class which will be used to populate the Starship. There are two source files:

- **Starship.java** which resides in the **com.spacex.vehicles** package
- **Astronaut.java**, which resides in the **com.spacex.personnel** package.

The main method for the Starship class is used as the application's main method; the program creates a Starship vehicle with an initial weight and altitude and adds astronauts. Each astronaut has a random height (cm) and weight (kg) within a specified range; when an astronaut is added, the total vehicle weight is updated with the astronaut's weight.

The source code files are provided in this exercise's GitHub repository.

New Application Features

Add a new instance variable to the Astronaut class which represents the time of arrival to the Starship vehicle. This variable must be declared as a **LocalDateTime** type.

Modify the overloaded Astronaut constructor to accept a LocalDateTime parameter which is used to set the new instance variable.

Modify the Astronaut toString method to display this value down to a one-second granularity.

In the addAstronaut method in the Starship class, declare a local LocalDateTime variable to set the time the astronaut is being added to the vehicle. Initialize this variable to the current date and time (e.g. **LocalDateTime ldtArrival = LocalDateTime.now();**)

Solution begins on the following page.

```java
// Starship.java
// D. Singletary
// 3/20/23
// class representing SpaceX Starship

package com.spacex.vehicles;

import com.spacex.personnel.Astronaut;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.text.DecimalFormat;

// "As the most powerful launch system ever developed, Starship will be able
// to carry up to 100 people on long-duration, interplanetary flights."
// https://www.spacex.com/vehicles/starship


public class Starship {
    private static int MAX_ASTRONAUTS = 100;
    private String name;
    private double weightKg;  // weight in kg
    private double altitudeKm; // altitude in kilometers
    ArrayList<Astronaut> astronauts = new ArrayList<>();

    public static void main(String[] args) {

        // create a space station and display it
        Starship ss =
            new Starship("SN15", 4989516.07 /* base weight: 11M lbs*/);
        System.out.println(ss);

        // add some astronauts
        System.out.println("Adding astronauts!\n");

        // Crew 1: https://blogs.nasa.gov/crew-1/
        // Crew 2: https://blogs.nasa.gov/crew-2/
        // Crew 3: https://blogs.nasa.gov/crew-3/
        // Crew 4: https://blogs.nasa.gov/crew-4/
        // Crew 5: https://blogs.nasa.gov/crew-5/
        // Crew-6: https://blogs.nasa.gov/crew-6/

        // "In general, astronauts should weigh between 50 and 95 kilograms
        // (110 and 209 pounds) and measure between 149.5cm and 190.5cm."
        // https://nypost.com/2022/02/13/what-does-it-take-to-be-a-nasa-
astronaut/
```

```
        final double MIN_WT = 50, MAX_WT = 95; // cm
        final double MIN_HT = 149.5, MAX_HT = 190.5; // cm

        ss.addAstronaut("Megan McArthur",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));
        ss.addAstronaut("Woody Hoburg",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));
        ss.addAstronaut("Andrey Fedyaev",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));
        ss.addAstronaut("Shannon Walker",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));
        ss.addAstronaut("Jessica Watkins",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));
        ss.addAstronaut("Koichi Wakata",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));
        ss.addAstronaut("Matthias Maurer",
              getRandomInRange(MIN_WT, MAX_WT), getRandomInRange(MIN_HT,
MAX_HT));

        // set the altitude (low-earth orbit max
        // https://www.nasa.gov/leo-economy/faqs)
        ss.setAltitude(2000.0);

        // show the new state
        System.out.println(ss);
    }

    // overload, create a space station object with specified name and weight
    public Starship(String name, double weightKg) {
        this.name = name;
        this.weightKg = weightKg;
        this.altitudeKm = 0;
    }

    public static double getRandomInRange(double min, double max) {
        return(Math.random() * (max - min) + min);
    }

    // add an astronaut
    public void addAstronaut(String name,
                double heightCm, double weightKg) {
        LocalDateTime ldtArrival = LocalDateTime.now();
        astronauts.add(new Astronaut(name, heightCm, weightKg, ldtArrival));
        this.weightKg += weightKg;
    }

    // set the altitude
    public void setAltitude(double altitudeKm) {
        this.altitudeKm = altitudeKm;
    }

    // show the space station
    public String toString() {
        DecimalFormat df = new DecimalFormat(".#");
        String ss = "Starship: " + name + "\n" +
               "Weight (kg):   " + df.format(weightKg) + "\n" +
               "Altitude (km): " + altitudeKm + "\n" +
               "Astronauts:    " + astronauts.size() + "\n" +
               ((astronauts.size() > 0) ? "Astronaut Details:\n\n" : "");

        for (Astronaut a : astronauts)
            ss += a + "\n";

        return ss;
    }
}
```

```java
// Astronaut.java
// D. Singletary
// 3/20/23
// class representing an astronaut

package com.spacex.personnel;

import java.time.LocalDateTime;
import java.text.DecimalFormat;
import java.time.format.DateTimeFormatter;

public class Astronaut {
    private String name;
    private double heightCm;  // height in centimeters
    private double weightKg;  // weight in kilograms
    private LocalDateTime arrival; // arrival date/time

    // overload, create an astronaut with specified name, height, and weight
    public Astronaut(String name, double heightCm, double weightKg,
                     LocalDateTime arrival) {
        this.name = name;
        this.heightCm = heightCm;
        this.weightKg = weightKg;
        this.arrival = arrival;
    }

    @Override
    public String toString() {
        DecimalFormat decFormatter = new DecimalFormat(".#");
        DateTimeFormatter dtFormatter =
            DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");

        return "name: " + name + "\n" +
               "height: " + decFormatter.format(heightCm) + " (cm)\n" +
               "weight: " + decFormatter.format(weightKg) + " (kg)\n" +
               "arrival: " + dtFormatter.format(arrival) + "\n";
    }
}
```