



UNIVERSITY OF SRI JAYEWARDENEPURA

B.Sc. (General) Degree First Year

Second Semester Terminal Course Unit Examination – June 2021

CSC 111 1.0 Computer Programming Laboratory

Duration: 3 hours

Submission Deadline: 6:00 PM on 10th June 2021

Instructions:

- Use Java programming language to do the task.
- Your code should be formatted appropriately and described with comments where it is necessary.
- This is an individual task and will be checked for plagiarism.
- Found guilty of plagiarism may cause a penalty of up to 100% of mark deduction.
- Clearly state any assumption you made. Your assumptions must be fair and should not conflict with the existing facts in this document.
- Deadline: 6 PM 10th June 2021

Description:

Hydroponic farming is a method of growing plants using mineral nutrient solutions, in water, without soil in a greenhouse. However, several factors should be monitored and controlled to obtain a reasonable crop consistently over the year. A project has been initialized for monitoring and controlling these factors remotely. Your task is to develop the following modules of the software by using the given code at the end of this document.

Module 1: Plant profile

[60 Marks]

The plant profile module of the software is responsible for reading, storing, and providing the details of plants, plant batches, and their crop harvest that are to be used within the system. A batch is a single season of a set of plants that have been put in the greenhouse. When the plants in a batch reach the end of their lifetime, a new batch will be planted. A plant has the following attributes.

- Plant name
- Growing duration (in number of days)
- Cropping interval (in number of days)
- Lifetime (in number of days)

There are two main types of plants namely fruits and vegetables. The system needs to keep the following details of the two types.

- Fruit:
 - Standard volume (in cubic millimeter)
 - Standard Average Color (0 - 16,777,215)
 - Standard pH (0-14)
- Vegetables:
 - Standard area per leaf (in square millimeter)
 - Standard thickness (in micrometers)

A greenhouse typically cultivates several fruits and vegetables per batch. Only one batch is in operation in a greenhouse at any given time. A batch has the following details.

- Batch number (a whole number that is increased by one with planting a new batch)
- List of plants
- Date of planted (day, month, year)
- End date of batch (the date after elapsing the lifetime of the plant with the longest lifespan, from the date of planting.)

Each plant can have zero or more crop harvests. A crop contains the following details:

- Date of harvest
- Total Weight (in Kilogram)
- State (one of **Standard/Below Standard/Above Standard**)

You should create the class structure for the above module and write a driver program to test the module. The driver program should contain the following features:

- Enter the details of a batch of plants.
- Select a plant from the batch and do the following.
 - View the plant details.
 - View the details of all crops.
 - Add a crop harvest.
 - Check whether the maximum crops are reached (use crop interval and lifetime of the plant to obtain this)

Module 2: Sensor Monitor

[40 Marks]

A simulator of a generic sensor is provided in this document. There are three main sensors in the greenhouse namely,

- temperature
- humidity
- pH

Use the sensor simulating class given at the end of this document to create the three sensors. Use the given interface to create a program that prints the simulated sensor values in the following format on the screen as soon as the sensors measured a new value.

Sensor Name: *sensor_type*, Measure: *measured_value*

Ex:- Sensor Name: TEMPERATURE, Measure: 0.5

Sensor simulating class and the listening interface:

```
import java.util.*;
interface SensorListener{
    public void onValueUpdated(SensorTypes type, double value);
}

enum SensorTypes
{
    TEMPERATURE, PH, HUMIDITY;
}

class Sensor extends Thread{
    private SensorTypes type;
    private Random rnd;
    private SensorListener listener;

    public void setSensorListener(SensorListener listener){
        this.listener = listener;
    }

    public void setType(SensorTypes type){
        this.type = type;
    }

    public SensorTypes getType(){
        return this.type;
    }

    public Sensor(SensorTypes type){
        this.type = type;
        rnd = new Random();
    }

    public void startSensor(){
        start();
    }

    public void run(){
        while(true){
            try {
                if(listener != null){
                    listener.onValueUpdated(type, rnd.nextDouble());
                }
                Thread.sleep(1000);
            }catch (InterruptedException e) {
                System.out.println("thread interrupted");
            }
        }
    }
}
```