

Temperature has a colour

Design and construct a computer program for the Arduino Microprocessor that monitors temperature sensor input, changing the colour of an RGB LED.

Achieved	Merit	Excellence
Construct a advanced computer program	Skillfully construct a advanced computer program	Accurately construct a advanced computer program

Introduction

This assessment requires students to design and construct a computer program for the Arduino Microprocessor that monitors temperature sensor input, changing the colour of an RGB LED to match current temperature .Students must demonstrate the use of advanced iterative processes to design, develop and test their program to ensure they construct an outcome that meets end user specifications.

This Assessment starts on XXX and Finishes XXX . This is 3 weeks (approx 12 in class hours)

Specifications

- Tact switch used to set state of system ??? add more here ?/
- Store input from an analogue temperature sensor
- RGB LED colour corresponds to temperature (see diagram)

Assessment Notes [Teacher copy]

Program will use INT and BOOLEAN data types

Program will use conditional statements for change of state

Program will use FOR loop to switch each of the RGB Pins HIGH, LOW

Program will use an ARRAY to hold the RGB pins

Program will use setup(), loop() functions with no parameters

Program will use an led_colour(x) function which takes temp as parameter

Program will use a current_temp() function which returns value for use. With this function 5 samples input samples are taken and averaged.<https://learn.adafruit.com/thermistor/using-a-thermistor>

Program will use WHILE to validate inputs



What to do next?

Use the following as a guide in the design and construction of your Computer Program

Planning the Inputs, Processes and outputs

1. What inputs are required from the user?
2. How will you store your data? What names will you use for variables and lists?
3. What calculations and decisions are required?
4. What output does the program require?

Plan the testing

Make a testing table and include in the inputs you plan to use for testing your program. Think about how you will test the program for various cases such as expected, boundary and unexpected input

Inputs	What should happen?	What happened when program is executed?

Develop the program

Create the program in an iterative manner.

- Start by breaking down the program into smaller achievable parts then code.
- Test, debug and evaluate each part before moving on the next progressive improvement.
- Save each version of the code as you make progress and include a copy of your code at each step in your portfolio
- Document the program with a comment block at the top and code comments
- Make the program
 - flexible using meaningful variables names, constants and lists
 - robust by validating input before use

Testing on expected, boundary and unexpected input

Complete testing of your program

Sample Code for Achieved <https://repl.it/MbnQ/1>

```
# Community Nursery Program
# demo program, version 1

# Welcome
print("Te Kakeno Community Nursery Project Quote Application")

# Get all required input. Note there is no validation on input
# .. do this validation with a function
area = float(input("What is the approximate area of land in meters squared? "))
density = float(input("How many plants do you expect to plant in a typical meter square? "))
location = input("Is the location of the planting in Wanaka? (Y/N) ").lower().strip()

# assign plant types to a list
print("For each of the plants shown, select 'Y' to include or 'N' to remove")
plantTypes = ['Flax', 'Lancewood', 'Kanuka', 'Koromiko', 'Mikimiki']
plantsForProject = []

# determine what plants the project requires
for plant in plantTypes:
    doRequire = input(plant + " Y/N ").lower().strip()
    if doRequire=="y":
        plantsForProject.append(plant)

# Determine || Assign Equipment and transport Costs
if location == "n":
    transportCost = 100
else:
    transportCost = 0

# Some constants
equipmentCost = 200
averagePerUnitCost = 6.50

# A calculation of final cost on the project
TotalCost = ( area * density * averagePerUnitCost ) + transportCost + equipmentCost

# output required information to the user
print("Planting requirements include:", len(plantsForProject), "Native plant types")
print(*plantsForProject, sep=',')
print("Total plants required", (area * density) )
print("The Average Cost per plant is:", averagePerUnitCost)
print("The Total Cost for the project is", TotalCost)
```