COMPUTER SCIENCE 1A PRACTICAL 3 DESIGN

Problem Description

Design a system for The Utopian Electrical Supply Commission as it is in need of a system to forecast the likelihood of load-shedding based on the percentage of unplanned outages at each of its three power stations. Not every power station is equally crucial to the stability of the grid, so a weighted average is needed to calculate the overall likelihood. Any errors encountered during the running of this program must reported. The system must prompt the user (send useful messages telling them what to do).it must make use of unique constant error codes (1 per type of error). Range of the percentages [0 –100] must be enforced so that the user input values as percentages in the correct range. The system ensure that conversion errors are detected and that the user is given one chance to retry. Users must input the weights and they must be enforced to be in the range of [0.0 –1.0]. the sum of the weights must equal to 1.0.Then lastly load shedding likelihood score is calculated as a weighted average and Based on the score an alert level is assigned based on the following ranges [0 –59]–Low,[60–79] –Medium◦[80 –100] –High, Anything else –"Error"

Input & Output

Input	
Input Description	*Mechanism*
Percentage of an unplanned outage for station1 (Non-negative [0-100], Whole Value)	**S**tandard input Stream
Percentage of an unplanned outage for station2 (Non-negative [0-100], whole Value)	**S**tandard input Stream
Percentage of an unplanned outage for station3 (Non-negative [0-100], whole Value)	**S**tandard input Stream
Weight of an unplanned outage for station1 (non-negative [0.0-1.0], Decimal Double Value)	**S**tandard input Stream
Weight of an unplanned outage for station2 (non-negative [0.0-1.0], decimal Double Value)	**S**tandard input Stream
Weight of an unplanned outage for station3 (non-negative [0.0-1.0], Whole Double Value)	**S**tandard input Stream
Double Value for the sum of weights (non-negative, sum to one)	Compute from summing the weights
Double Value for the weighted average (non-negative)	Computed from summing the products of each percentage and weight
Output	
Character Data type for indicating the high, low, or medium is the possible outage.	Standard output (Get its value from a if else statement)

Data format

dentifier	*Data Type*	*Description*
dblSt1	Double unsigned	Stores percentage outage for station 1
dblSt2	Double unsigned	Stores percentage outage for station 2
dblSt3	Double unsigned	Stores percentage outage for station 3
dblWSt1	Double unsigned	Stores weight of the outage for station 1
dblWSt2	Double unsigned	Stores weight outage for station 2
dblWSt3	Double unsigned	Stores weight outage for station 3
dblsum	Double unsigned	Stores sum for weights
dblAvg	Double unsigned	Stores sum for a weighted average
chrAlert	character	Stores the level of alert for a possible outage

Pseudo Code

```
Let dblSt1, dblSt1, dblSt1 = 0.0
Prompt user “enter percentage for station1”
Read --> dblSt1
If cin fails then
        Let bin <--” empty string”
        Clear cin
        Read --> “gabbage”
        Read --> dblSt1
        If cin fails again then
                Print “fatal error”
                Exit
If dblSt1 < 0 or > 100
        Print error <-- out of the range
        Read --> dblSt1
        If dblSt1 < 0 or > 100
                Print error <-- out of the range
                exit

Prompt user “enter percentage for station2”
Read --> dblSt2
If cin fails then
        Let bin <--” empty string”
        Clear cin
        Read --> “garbage”
        Read --> dblSt2
```

```
        If cin fails again then
                Print “fatal error”
                Exit
If dblSt2 < 0  or > 100
        Print error <-- out of the range
        Read --> dblSt2
        If dblSt1 < 0  or > 100
                Print error <-- out of the range
                exit


Prompt user “enter percentage for station3”
Read --> dblSt3
If cin fails then
        Let bin <--” empty string”
        Clear cin
        Read --> “garbage”
        Read --> dblSt3
        If cin fails again then
                Print “fatal error”
                Exit
If dblSt2 < 0 or > 100
        Print error <-- out of the range
        Read --> dblSt3
        If dblSt1 < 0 or > 100
                Print error <-- out of the range
                exit

Let dblWst1, dblWst1, dblWst1 = 0.0
Prompt user “enter weight for station1”
Read --> dblWst1
If cin fails then
        Let bin <--” empty string”
        Clear cin
        Read --> “garbage”
        Read --> dblWst1
        If cin fails again then
                Print “fatal error”
                Exit
If dblWst1 < 0.0or > 1.0
        Print error <-- out of the range
        Read --> dblWst1
        If dblSt1 < 0.o or > 1.0
                Print error <-- out of the range
                Exit

Prompt user “enter weight for station2”
Read --> dblWst2
```

```
If cin fails then
        Let bin <--" empty string"
        Clear cin
        Read --> "garbage"
        Read --> dblWst2
        If cin fails again then
            Print "fatal error"
            Exit
If dblWst12< 0.0or > 1.0
        Print error <-- out of the range
        Read --> dblWst2
        If dblSt2 < 0.o or > 1.0
            Print error <-- out of the range
            exit

Prompt user "enter weight for station3"
Read --> dblWst3
If cin fails then
        Let bin <--" empty string"
        Clear cin
        Read --> "garbage"
        Read --> dblWst3
        If cin fails again then
            Print "fatal error"
            Exit
If dblWst3 < 0.0or > 1.0
        Print error <-- out of the range
        Read --> dblWst3
        If dblSt3 < 0.o or > 1.0
            Print error <-- out of the range
            exit

Let chrAlert
Let dblAvg <-- dblWst1* dblSt3 + dblWst2 * dblSt3 +  dblWst3 * dblSt3
Let dblSum <--  dblSt3 +  dblSt3 +  dblSt3

If dblAvg >= 0 and <=59 then
        ChrAlert <-- L

If dblAvg >= 60 and <=79 then
        ChrAlert <-- M

If dblAvg >= 80 and <=100 then
        ChrAlert <-- M

Switch on chrAlert
    Case L
        Print Low
```

Switch on chrAlert
    Case M
        Print Mediam

Switch on chrAlert
    Case H
        Print High

UML Activity Diagram