

## BCI433 Lab 3B (Summer 2018)

### Writing an interactive RPGLE screen program

#### Lab objectives:

- Use Case structure to solve an application problem
- Use the Debug option to check a program's logic

#### Lab Requirements:

Hand in the compile listing for HWYTOLLRPG (with toll logic)

Successfully run HWYTOLLRPG (with toll logic)

DSPLIBL

Start an RDi session

Start a 'Green Screen' (emulator) Session.

#### Using Rational Developer for i (RDi):

#### Part A

##### Objectives:

SRCPF: LIB433LIB/LAB3SAMPLE:

Member: DAYSRPG.rpgle

**Case structure** is a good choice to solve application logic problems. Here is an example of that:

```
SELECT;
When Day = 1;
    DayName = 'Monday';
When Day = 2;
    DayName = 'Tuesday';
When Day = 3;
    DayName = 'Wednesday';
When Day = 4;
    DayName = 'Thursday';
```

```
.....
ENDSL;
```

If Day is a 1, then the first test would be done, the DayName would be set to 'Monday' and control would go to ENDSL. The test for Day = 2 would not be performed. If Day is a 7 then seven tests would be done and a DayName = 'Sunday' statement would be executed.

If an 8 or 9 was entered in Day you could include an Other clause in place of a When clause at the bottom of this structure before the ENDSL. Other is a catchall.

```
Other ;
    DayName = 'Invalid Day Number' ;
ENDSL ;
```

Logical Operators:

Tests done with the select When statement can include <=, >=, <> (not equal to), AND, OR

In class **exercise**:

**You need to figure out how to convert a number to a grade.**

Tests are entered as a mark out of 100 and the average of TEST1 and TEST2 (TESTAVG) is worth 35%. Labs are worth 30% and the mark in LABS is a mark out of 30. The final exam EXAM is out of 100 and is worth 30% of the final grade. The final grade is stored in a field called NUMGRADE. After this grade has been determined, it needs to be converted to a letter grade LETGRADE.

The following chart is used to do this.

A+ = 90 - 100  
C+ = 65 - 69  
F = 0 - 49

A = 80 - 89  
C = 60 - 64

B+ = 75 - 79  
D+ = 55 - 59

B = 70 - 74  
D = 50 - 54

Solution:

```
SELECT;  
  WHEN (NUMGRADE >= 90);  
    LETGRADE = 'A+';  
  when (numgrade >= 80);  
    letgrade = 'A';  
  when (numgrade >= 75);  
    letgrade = 'B+';  
  when (numgrade >= 70);  
    letgrade = 'B';  
  when (numgrade >= 65);  
    letgrade = 'C+';  
  when (numgrade >= 60);  
    letgrade = 'C';  
  when (numgrade >= 55);  
    letgrade = 'D';  
  other;  
    letgrade = 'F';  
ENDSL
```

Do not test both the upper and lower limits of a range – you only need to test one or the other depending on your approach. Testing both limits is wasting CPU time. Not a big problem with this application, but may be a factor when processing millions of records.

You have an interactive highway toll RPGLE program running from lab 3a with no TOLL logic to determine the correct toll. Apply the information below to determine the correct tolls for completion of lab 3b.

Processing.

The first screen allows entry of number of kilometers traveled, the type of vehicle, the time of day, and whether or not the traveler has a transponder. Then the first screen and overlaid second screen are displayed. The amounts showing on the second screen include the toll rate, the toll charge based on the toll rate and number of kilometers travelled, a possible no transponder charge of \$4.07 if the person does not have a transponder (it is 0 if they do have a transponder) and the total charge for the trip. Values can be entered until user presses F3 when the second screen is showing. No validation is required in the program; for example the display file will not allow someone to enter X as the vehicle type.

**The fields used to contain the first screen values are protected from user input whenever both screens show.**

Determining the toll rate based on AdjustHour determined in AdjustTime subroutine

Time (HOUR)	Light Vehicle	Heavy Vehicle	Heavy Multiple Vehicle
6 am - 8:59 am	37.07 cents a km	74.18 cents/km	111.22 cents/km
9 am - 9:59 am	32.59 cents a km	65.18 cents/km	97.78 cents/km
10 am - 3:59 pm	28.33 cents a km	56.66 cents/km	84.99 cents/km
4 pm - 6:59 pm	38.42 cents a km	76.83 cents/km	115.25 cents/km
7 pm - 5:59 am	21.62 cents a km	43.24 cents/km	64.86 cents/km

AdjustTime: The mintues do not matter and are not entered. 8:00 is 8 and 8:59 is still 8.

The time is entered as an hour and whether it is AM or PM. If the hour is PM you need to add 12 to the hour in a field called AdjustHour. If the hour is AM then AdjustHour is the same as hour. If the hour is 12 and the time is AM you need to set AdjustHour to 0. If the hour is 12 PM you need to not add 12 to adjust hour. Use AdjustHour in figuring out the appropriate rate.

