

LAB OBJECTIVE

At the end of this lab activity, the students should be able to:

- Use struct to solve programming problems.

PRACTICE

1. Create a structure called *Area* with values *width*, *height* and *area* (float).
 - In the main() function:
 - Declare a structure variable array called *rectangle* with size 3.
 - Using a while loop, ask the user to enter values for width and height.
 - Call function *get_area(...)*, passing the *width* and *height* as parameter.
 - Using for loop, display the area value as shown below.
 - In function *get_area(...)*:
 - Calculate and return the area.

```
Enter width and height of rectangle 1 (in m): 4 5
Enter width and height of rectangle 2 (in m): 5 6
Enter width and height of rectangle 3 (in m): 6 10

Area of rectangle 1 with 4.00 by 5.00 width and height is 20.00 metre square
Area of rectangle 2 with 5.00 by 6.00 width and height is 30.00 metre square
Area of rectangle 3 with 6.00 by 10.00 width and height is 60.00 metre square
```

2. Modify the code from Question 1. Create a structure called *Area* with values *width*, *height* and *area* (float). Declare a structure variable array called *rectangle* with size 3.
 - In the main() function:
 - Call function *get_area(...)*, passing *rectangle* as parameter.
 - In function *get_area(...)*:
 - Using a while loop, ask the user to enter values for width and height.
 - Calculate the area.
 - Using for loop, display the area value (output is similar to question 1).

3. Write a complete C program that generates a company payroll list. The requirements of the program are:

Declare the following constants using pre-processor directive:

- EPF_RATE : Initialize to 0.11
- PART_TIME_RATE : Initialize to 8.5

Create a structure called *Employee*. The data members are; *name* (string), *type*, *hours* (integer), *salary*, *epf*, *tax*, *net* (float).

In *main()*:

- Declare structure variable array *info*, size 4.
 - Call function *get_details()*, passing *info* as parameter.
 - Call function *calc_net()*, passing *info* as parameter.
 - Call function *display()*, passing *info* as parameter.

In function *get_details()*:

- Using a *for-loop* :
 - Get input for employee *name*.
 - Get input for employee *type*. Employee type is either 1 (full time) or 2 (part time). If user input the wrong employee type, display error message **Wrong Input – Please Reenter**. Program must continually get input for employee type until the correct employee type is entered.
 - If employee *type* entered is 1 (full time), get input for employee *salary*. Else for part time employee, get input for employee *hours*.

In function *calc_net(...)*:

- Using a *for-loop* :
 - If employee *type* entered is 1 (full time), calculate employee *epf* [*salary* × EPF_RATE].
 - Else for part-time employee, assign value zero to employee *epf* and calculate *salary* [*hours* × PART_TIME_RATE].
 - Calculate employee *tax* based on *salary*:

Salary	Tax
Equals or above RM 10,000.00	20% of salary
Less than RM 10,000.00 but at least RM 8,000.00	15% of salary
Less than RM 8,000.00 but at least RM 5,000.00	10% of salary
Less than RM 5,000.00	No tax

- Calculate employee *net salary* [*salary* – *epf* – *tax*].

In function *display(...)*:

- Display title :: **PAYROLL LIST - NET SALARY** ::
- Using a *for-loop*, display employee's *name* and *net salary*.

Sample output:-

```
Satff's 1 Name      : Johan
1[Full time]  2[Part time] : 5
Wrong Input - Please ReEnter

1[Full time]  2[Part time] : 1
Monthly salary      : RM 12000

Satff's 2 Name      : Joseph
1[Full time]  2[Part time] : 1
Monthly salary      : RM 5500

Satff's 3 Name      : Jimmy
1[Full time]  2[Part time] : 1
Monthly salary      : RM 3000

Satff's 4 Name      : Jack
1[Full time]  2[Part time] : 2
Total hours worked   : 300

:: PAYROLL LIST - NET SALARY ::
-----
Johan   RM 8280.00
Joseph  RM 4345.00
Jimmy   RM 2670.00
Jack    RM 2550.00
```

4. Write a program to calculate the students' coursework mark and determine whether it's an A (mark $\geq 40/50$).

Create a structure named *Student* which contains *student ID* (string), *lab*, *quiz*, *assignment*, *midterm* and *total* (float). Declare a structure variable array called *coursework* with size 3

- In *main()* function:
 - Using for-loop ask user to enter the *student ID*, marks for *lab*, *quiz 1*, *quiz 2*, *assignment* and *midterm*. Repeat for 3 students.
 - For quiz, declare an array quiz with size 2; *quiz 1* & *quiz 2*. (nested for-loop)
 - Calculate mark for *quiz* (*quiz 1* + *quiz 2*).
 - Call function *get_assignment(...)*, passing the assignment mark.
 - Call function *get_midterm(...)*, passing the midterm mark.
 - Calculate *total* (*lab* + *quiz* + *assignment* + *midterm*).
 - Call function *display_report(...)*, passing the *coursework*.

- In *get_assignment (...)* function:
 - Calculate and return the value of assignment as 15% weightage ($assignment / 80 * 15$).
- In *get_midterm (...)* function:
 - Calculate and return the value of midterm in 15% weightage ($midterm / 50 * 15$).
- In the *display_report(...)* function:
 - Using *for loop*, display student who obtains coursework mark more or equal to 40.00 (maximum coursework mark is 50).
 - Also display the total number of students who achieved the above.

Sample output:

```

Enter student ID #1           : X107119
Enter your lab marks (10%)    : 8.5
Enter your quiz 1 marks (5%)  : 4.5
Enter your quiz 2 marks (5%)  : 5
Enter your assignment marks (80%) : 75.5
Enter your midterm marks (50%) : 45.5

Enter student ID #2           : X107227
Enter your lab marks (10%)    : 4
Enter your quiz 1 marks (5%)  : 1
Enter your quiz 2 marks (5%)  : 1.5
Enter your assignment marks (80%) : 45.5
Enter your midterm marks (50%) : 23.5

Enter student ID #3           : X107299
Enter your lab marks (10%)    : 10
Enter your quiz 1 marks (5%)  : 5
Enter your quiz 2 marks (5%)  : 5
Enter your assignment marks (80%) : 80
Enter your midterm marks (50%) : 50

-----
                COURSEWORK GRADE A
-----

Student ID      : X107119
Coursework mark : 45.81

Student ID      : X107299
Coursework mark : 50.00

Total number of students with grade A : 2

```

SUBMISSION

Write a complete program that calculates and display a person's financing summary when he/she decides to purchase a car via hire-purchase.

Create structure called *Calculator*. The members are; *price*, *rate*, *downPayment*, *financingAmount*, *profitAmount*, *totalFinancing*, *monthlyInstallment* (float), *years* (int) and *model* (string). Declare structure variable array called *financing* with size 3.

In *main()*

- Call function *get_input()*, passing *financing* as parameter.
- Call function *get_finance()*: passing *financing* as parameter.
- Call function *display ()*, passing *financing* as parameter.

In function *get_input(...)*:

- Using a *for-loop*, prompt the user for the following inputs and store them into array *financing*. Repeat for 3 times.
 - *car model, price, down payment, financing rate and financing years.*

In function *get_finance(...)*:

- Using a *for-loop*, calculate the following and store the results into array *financing*:
 - *financing amount = car price – down payment*
 - *profit amount = financing amount × years × (rate ÷ 100)*
 - *total amount = financing amount + profit amount*
 - *monthly installment = total amount ÷ (years × 12)*

In function *display(...)*:

- Using a *for-loop*, display the following from array *financing*:
 - *car model, financing amount, profit amount, total amount and monthly installment.*

Sample output:

```

=====
=      CAR PAYMENT FORMULA      =
=====

Enter car model #1           : Honda BRV
Enter car price #1           : 85800
Enter down payment amount    : 8580
Enter financing rate          : 3.1
Enter number of financing years : 5

Enter car model #2           : Honda Jazz
Enter car price #2           : 73800
Enter down payment amount    : 7380
Enter financing rate          : 3.5
Enter number of financing years : 7

Enter car model #3           : Honda City
Enter car price #3           : 76100
Enter down payment amount    : 7610
Enter financing rate          : 3.2
Enter number of financing years : 7

=====
=      FINANCING SUMMARY      =
=====

Car #1
-----
Car model           : Honda BRV
Financing amount    : RM 77220.00
Bank's profit amount : RM 11969.10
Total amount        : RM 89189.10
Monthly installment : RM 1486.48

Car #2
-----
Car model           : Honda Jazz
Financing amount    : RM 66420.00
Bank's profit amount : RM 16272.90
Total amount        : RM 82692.90
Monthly installment : RM 984.44

Car #3
-----
Car model           : Honda City
Financing amount    : RM 68490.00
Bank's profit amount : RM 15341.76
Total amount        : RM 83831.76
Monthly installment : RM 998.00

```