SCHOOL OF COMPUTER SCIENCES
UNIVERSITI SAINS MALAYSIA
Semester II Session 2023/2024

Tutorial CPT113
Topic: Struct and Enum

Learning Outcomes:

- Understand and apply the use of struct and enum
- Convert from parallel array into struct
- Understanding the difference between different types of struct variable types / instances

Fundamental

1. Build a struct called `Student` with the following information:

   `Name`
   `ID`
   `Desasiswa`
   `Year`
   `Semester`
   `CGPA`

   a) Create a struct and accept input from user so that it can print back to the terminal.
   b) Define an array `stuList` that has 5 elements. Each element representing a student info. Each element must be read from file input given and the program must be able to display data stored in each element to the terminal.
   c) Modify your answer in (b) to accept the input from terminal
   d) Modify your answer in (c) above using pointer of structure

2. Given the following program:

```
#include <iostream>
#include <iomanip>
using namespace std;

enum Day { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY };

int main()
{
   const int NUM_DAYS = 5;    // The number of days
   double sales[NUM_DAYS];    // To hold sales for each day
   double total = 0.0;        // Accumulator
   Day workDay;               // Loop counter

   // Get the sales for each day.
   for (workDay = MONDAY; workDay <= FRIDAY;
                          workDay = static_cast<Day>(workDay + 1))
   {
      cout << "Enter the sales for day "
           << workDay << ": ";
      cin >> sales[workDay];
   }
```

```

      cin >> sales[workDay];
   }

   // Calcualte the total sales.
   for (workDay = MONDAY; workDay <= FRIDAY;
                          workDay = static_cast<Day>(workDay + 1))
      total += sales[workDay];

   // Display the total.
   cout << "The total sales are $" << setprecision(2)

   return 0;
}
```

The output should look something like this:

```
Enter the sales for day MONDAY: 100
Enter the sales for day TUESDAY: 200
Enter the sales for day WEDNESDAY: 300
Enter the sales for day THURSDAY: 400
Enter the sales for day FRIDAY: 500
The total sales are $1500.00
```

Modify the above program so the output can display the day instead of just the enumerated number.

3. Given the following parallel array, reconstruct the program to use stuct:
   Name:

San	Mukuda	Tamaru	Sonoko

   Skill:

Heal	Gourmet	Sword	Imagine

   Write appropriate set and get functions. Complete the main function to allow use to enter different values.

Applied

4. Write a program that uses a structure named `MovieData` to store the following information about a movie:
   Title
   Director
   Year Released
   Running Time (in minutes)

   The program should create two `MovieData` variables, store values in their members, and pass each one, in turn, to a function that displays the information about the movie in a clearly formatted manner.
   Reminder: Do not use array of instance /variable structure

5. Write a program with structure `Rectangle` and creating 3 structure instance named `kitchen`, `bedroom` and `den`. Calculate the total area of the three rooms. Measure, it should have a
   a) structure instance
   b) display the total area of three room.

6. Write a program that uses a structure to store the following weather data for a particular month:

   ```
   Total Rainfall
   High Temperature
   Low Temperature
   Average Temperature
   ```

   The program should have an array of 12 structures to hold weather data for an entire year. when the program runs, it should ask the user to enter data for each month. (The average temperature should be calculated.) Once the data are entered for all the months, the program should calculate and display the average monthly rainfall, the total rainfall for the year, the highest and lowest temperatures for the year (and the months they occurred in), and the average of all the monthly average temperatures.
   Input Validation: Only accept temperatures within the range between –100 and +140 degrees Fahrenheit.

7. Write a program that keeps track of a speakers' bureau. The program should use a structure to store the following data about a speaker:
   Name
   Telephone Number
   Speaking Topic
   Fee Required

   The program should use an array of at least 10 structures. It should let the user enter data into the array, change the contents of any element, and display all the data stored in the array. The program should have a menu-driven user interface.

   Input Validation: When the data for a new speaker is entered, be sure the user enters data for all the fields. No negative amounts should be entered for a speaker's fee.

8. Write a program that simulates inventory bins in a warehouse. Each bin holds a number of the same type of parts. The program should use a structure that keeps the following data:
   Description of the part kept in the bin
   Number of parts in the bin
   The program should have an array of 10 bins, initialized with the following data:

Part Description	Number of Parts in the Bin
Valve	10
Bearing	5
Bushing	15
Coupling	21
Flange	7
Gear	5
Gear Housing	5
Vacuum Gripper	25
Cable	18
Rod	12

The program should have the following functions:
`AddParts`—increases a specific bin's part count by a specified number.
`RemoveParts`—decreases a specific bin's part count by a specified number.

When the program runs, it should repeat a loop that performs the following steps:
The user should see a list of what each bin holds and how many parts are in each bin. The user can choose to either quit the program or select a bin. When a bin is selected, the user can either add parts to it or remove parts from it. The loop then repeats, showing the updated bin data on the screen.

Input Validation: No bin can hold more than 30 parts, so don't let the user add more than a bin can hold. Also, don't accept negative values for the number of parts being added or removed.

9. Write a program that has a dateTime structure to store the following particulars about date and time:
   Year
   Month
   Date
   Hours
   Minutes
   Seconds

   The program should declare two `dateTime` variables of the above structure and ask the user to input data on the date and time of two different instances. The program should pass these structure variables as arguments to a function. The function should determine the date (and not time) that comes earlier between the two arguments. If both dates are the same, the function displays a message "Identical dates" and returns any of the arguments; otherwise it should return the date that comes earlier.

   Input Validation: Do not accept negative values for year. Accept values for months only within the range 1-12. Accept hours between 0 and 23, minute between 0 and 59 and seconds between 0 and 59.

10. Write a program that calculates pay for either an hourly paid worker or a salaried worker. Hourly paid workers are paid their hourly pay rate times the number of hours worked. Salaried workers are paid their regular salary plus any bonus they may have earned. The program should declare two structures for the following data:
    Hourly Paid:
    HoursWorked
    HourlyRate
    Salaried:
    Salary
    Bonus

    The program should also declare a union with two members. Each member should be a structure variable: one for the hourly paid worker and another for the salaried worker. The program should ask the user whether he or she is calculating the pay for an hourly paid worker or a salaried worker. Regardless of which the user selects, the appropriate members of the union will be used to store the data that will be used to calculate the pay.

    Input Validation: Do not accept negative numbers. Do not accept values greater than 80 for HoursWorked.