TUTORIAL 8 (SECJ1013)

PROGRAMMING TECHNIQUE 1

SECTION 04 & 07, SEM 1, 2024/2025

Q#1

Table 1 indicates how the record of a patient is stored. Use the information provided to answer the questions in parts (a), (b) and (c).

Table 1

Field	Field Name	Field Data Type
Patient's name	**name**	string or array of 30 characters
Patient's doctor	**doctor**	string or array of 20 characters
Patient's ward	**pWard**	A structure type named **Ward** containing 2 fields: • Any integer between 1 – 5 to indicate floor • A character F or M to indicate gender eithera female or male ward, respectively.
Admission date	**admitDate**	A structure type named **Date** containing 3fields: • Any integer between 1 – 31 to indicate theday of admission • Any integer between 1 – 12 to indicate themonth of admission • Any integer to indicate the year of admission

a) Write C++ statements to declare all the data types required for storing the record of a patient. Name the data type for a patient's record as **PatientRecord**.

b) Declare a variable called **pRec** to store records for 50 patients. Set the twentieth patient's name to "Robert Kwok" and the ward to a male ward located on the fifth floor.

c) Write the statements to perform each of the following tasks:

i) Read in from the keyboard the thirty-fifth patient's doctor and admission date.

ii) Find the total number of male patients and the total number of female patients found on the second floor.

iii) If the last patient's name is "Mohammed", print out the patient's doctor and his/her ward information.

Q#2

Given the declaration for a structured data type named **S_TYPE** as follows.

```
struct S_TYPE
{
   float fixedBefore;
   CHOICE choice;
   U_TYPE flexible;
   float fixedAfter;
};
```

Declare further:

a) the data type **CHOICE** which is an enumerated data type whose members are **ONE** and **TWO**.

b) the data type **U_TYPE** which is a union data type which can be made up of either **choice1** a character data type or **choice2** an integer data type.

Q#3

A structure type called **SalesRecord** is designed to hold all necessary data regarding a salesman in a company and the sales information that he or she has made. The **SalesRecord** structure and an array variable called **salesRep** are declared below:

```
struct SalesRecord
{
   int id;
   char name[30];
   double quarterlySales[4];
   double totalAnnual;
};

SalesRecord salesRep[20];
```

a) A function named **readRecord** is designed to read the information of all the salesmenof the company from the user. Assume that the company has 20 salesmen. Below is the prototype of the function.

   void readRecord (SalesRecord []);

   i) Write a complete definition for the function **readRecord** to read the salesman's id, name and quarterly sales.

   ii) Write a proper call statement to the function **readRecord** to fill in the array **salesRep.**

b) The member **totalAnnual** of **SalesRecord** is used to hold the information about the total amount of yearly sales of each salesman respectively. The calculation of the total sale is accomplished by the following function:

```
double getAnnualSale (double quarterlySales[])
{
   double total = 0.0;
   for (int i=0; i< 4; i++)
   total += quarterlySales[i];
   return total;
}
```

Write a call statement to this function to calculate the annual sale for every salesman in the list.

c) A function named **getSalesman** is used to search for the record of a salesman using the salesman's id from the salesmen list of record. The following program segment is written to call to this function. Define the function **getSalesman** in order to accomplish its purpose.

```
:
SalesRecord searchRecord:
int id;
:
cout << "Enter Salesman's ID =>" << endl; cin >> id;
searchRecord = getSalesman(salesRep, id);
:
:
```

Q#4

Complete the following program with an appropriate C++ statements. This program stores the information of five (5) lines using structures and determines the length of each line. Declare a structure to represent point with two (2) members of type **float**; x-coordinate and y-coordinate. Then, create a structure named **Line** with three (3) members. The members include length of type **float**, point A and B. Store five (5) input data in an array named **ln** of type **Line**. Formula for length is $Length = \sqrt{(X_B - X_a)^2 + (Y_B - Y_a)^2}$.

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

int main()
{
   /*a) Declare structure named Point with 2 members of type float
   named x and y*/
     ___________________a(i)____________________  {
     ___________________a(ii)___________________
     ___________________a(iii)__________________
   };

   /*b) Declare structure named Line with 2 members of type Point
   named A and B, and one member of type float named length*/
     ___________________b(i)____________________  {
     ___________________b(ii)___________________
     ___________________b(iii)__________________
     ___________________b(iv)___________________
   };

   const int SIZE = 5;

   //c) Define array named ln of type Line with size 5
   ______________________________________________

  //Get input data
   for (int i = 0; i < SIZE; i++) {
     cout << "LINE - " << i+1 << endl;
     cout << "Enter x-coordinate and y-coordinate of point A: ";
     //d) Get input for point A
     ______________________________________________

     cout << "Enter x-coordinate and y-coordinate of point B: ";
     //e) Get input for point B
     ______________________________________________
   }

  //Calculate length of line
   for (int i = 0; i < SIZE; i++) {
     float diffX, diffY;

     //f) Calculate difference in x-coordinate for 2 points
     ______________________________________________

     //g) Calculate difference in y-coordinate for 2 points
     ______________________________________________

     //h) Calculate length between 2 points
     ______________________________________________

     cout << "Length of line - " << i+1 << " is ";

     //i) Display length
     ______________________________________________
   }

  return 0;
}
```

Q#5

After executing the following program segment, what is the value of the variables and output of the program? Write your answer in **Table 5**.

```
struct Pixel {int C, R;};
union value {int m, n;};
enum color {black, blue, red};

Pixel X = {40, 50}, Y, Z;
value val;
color obj = blue;
Z = X;
X.C += 10;
Y = Z;
Y.C += 10;
Y.R += 20;
Z.C -= 15;
val.m = X.R;
val.n = (int)obj + 1;

cout << obj << endl;
cout << val.m << "," << val.n;
```

Table 5

X		Y		Z		OUTPUT
C	R	C	R	C	R	

Q#6

a) Write a C++ code segment to define a structure type named **`Student`**, with the following members:

`name`	: an array of characters
`weight`	: a float value
`height`	: a float value

b) Write a C++ statement(s) to define a global constant named **`NUM`** that equal to 5 (the number of students).

c) Write a C++ statement(s) to define a global array of structure type **`Student`** named **`stud`** with size **NUM**.

d) Define a function named **`getData`** to read five (5) students' data (**`name`**, **`weight`** and **`height`**) from user input and insert it into the array that defined in (c). The function will not have any return value or received any value from **`main()`** function.

Q#7

Program 1 below is meant to ask the user to enter a diameter of circle, and display the radius and area of the circle. The program uses a function to return a structure. The area of the circle is expressed by the formula $A = \pi r^2$, and the radius of the circle is expressed by the formula *radius = diameter/2*. In function `calculateRadius`, user enters the diameter of the circle and then calculates the radius.

Complete **Program 1**, based on the instructions or comments written in (a) to (g).

Line	Program 1
1	`#include <iostream>`
2	`#include <cmath>`
3	`using namespace std;`
4	
5	`const struct double PI = 3.14159;`
6	
7	`struct Circle`
8	`{`

```
   double radius;
   double diameter;
   double area;
};

// (a) Write function prototype for calculating radius.
  ________________________________________;

int main(){
// (b) Declare a structure variable for circle
  ________________________________________;

   c = calculateRadius();

// (c) Write statement to calculate the circle’s area using
//     suitable predefined function.

  ________________________________________________;


// (d) Write statement to display the circle data.
  ___________________________________________;
  ____________________________________________;
   return 0;
   } // end of main function

// Function to calculate radius
  Circle calculateRadius()
  {
//(e) Declare a temporary structure variable for circle.

   ______________________________________________;
//(f) Write statement to get input for diameter of circle.
   cout << “Enter the diameter of circle: “;

   ________________________________________________;
//(g) Write statement to calculate radius of circle.

   ________________________________________________;
   return tempCircle;

   } // end of calculateRadius function
```