Lab 05- STRUCTURE

Structure:
C++ arrays allow you to define variables that combine several data items of the same kind, but **structure** is another user defined data type which allows you to combine data items of different kinds.

Structures are used to represent a record, suppose you want to keep track of your books in a library. You might want to track the following attributes about each book −

- Title
- Author
- Subject
- Book ID

For example:

```
struct Books {
   char  title[50];
   char  author[50];
   char  subject[100];
   int   book_id;
}
```

Types of Members
Structures in C++ can contain two types of members:

- **Data Member:** These members are normal C++ variables. We can create a structure with variables of different data types in C++.
- **Member Functions:** These members are normal C++ functions. Along with variables, we can also include functions inside a structure declaration.

Access structure member:
To access any member of a structure, we use the **member access operator (.)**. The member access operator is coded as a period between the structure variable name and the structure member that we wish to access. You would use **struct** keyword to define variables of structure type. Following is the example to explain usage of structure.

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

struct Books {
   char  title[50];
   char  author[50];
   char  subject[100];
   int   book_id;
};

int main() {
   struct Books Book1;       // Declare Book1 of type Book
   struct Books Book2;       // Declare Book2 of type Book
  cin>>Book1.title;
  cin>>Book1.author;
  cin>>Book1.subject;
```

Access modifier:

There are 3 types of access modifiers available in C++:
1. Public
2. Private
3. Protected

Example:

```
struct s1
{
int a; // public
private:
int b; // private
protected:
int c; // protected
public:
int d; // public again
};
```

Problem 1:

Write a structure to store the roll no., name, age (between 11 to 14) and address of students (more than 10). Store the information of the students.
1 - Write a function to print the names of all the students having age 14.
2 - Write another function to print the names of all the students having even roll no.
3 - Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).

Problem 2:

Enter the marks of 5 students in Chemistry, Mathematics and Physics (each out of 100) using a structure named Marks having elements roll no., name, chem_marks, maths_marks and phy_marks and then display the percentage of each student.

Problem 3:

Write a program to take the differences from start and stop time using structures. There should be proper adjustment for seconds, minutes and hours.

```
struct TIME {
int hours;
int minutes;
int seconds;
struct TIME differenceBetweenTimePeriod(struct TIME start, struct TIME stop)
{ }
};
```

Problem 4:

Write a program to add two distances in inch-feet using structure. The function designed for sum should be inside the structure and inch must be added to inch and similar for feet. If inches exceed from 11 and reaches to 12 it will be a feet then there should be increment in feet sum and decrement in inch and leverage inches less than 12.The return type of the function should be struct. The function name should be “add” and the return type must be a struct; the function definition must be inside the structure.
(Hint: 1 feet=12 inches)