You may be asked to demonstrate/explain your work to the tutor, if you are absent/unavailable or fail to demonstrate properly, zero marks will be awarded.

Exercises:

1. Explain the below with an example (an example should be a complete c program with a screenshot of the output)
   a. Function prototypes
   b. Typedef's
   c. Typecasting
   d. Structures
   e. Self-referential structures
   f. malloc() function
   g. realloc () function

2. Explain the differences between
   a. Arrays and Structures
   b. Structures and Self-referential structures
   c. User defined datatypes and system defined datatypes

After implementing functions, you have to write a complete C program.

3. Declare a structure and then write a main(),inside the main store the structure in a local variable. main() will call a function PrintData() by Passing this local variable and print the values of structure in PrintData().

   For question 4 use the following structure

```
// structure definition
struct Date
{
   int yy, mm, dd;
};


struct Emp
{
   char EmpName[25];
   float Salary;
   struct Date hired;
};
```

4.

a) Write main(), which will define an array called a[5] of type struct Emp. main() will then call GetData() in a loop 5 times, each time getting the data in the proper slot of the array. Then main() will call PrintData() in a loop so that the data is printed in reverse order.

```
// function prototypes
struct Emp GetData();
void PrintData(struct Emp);
```

b) Write a function that will receive one array of type struct Emp and an integer that gives the size of the array. Remember that, if the size is 5, then the largest index of the array should be 4. It will then return those employees' average salary.

```
// function prototypes
struct Emp GetData();
double GetAverage(struct Emp [], int);
```

c) Write a function that will receive one array of type struct Emp and name of an employee. The function should return the salary of that employee.

```
// function prototypes
struct Emp GetData();
float findsalary(struct Emp x[],char p[]);
```