**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **19/06/2020** | **Name:** | **Lepakshi T V** |
| **Course:** | **C Learning** | **USN:** | **4AL17EC044** |
| **Topic:** | * **Structure and unions** * **Memory management** | **Semester & Section:** | **6th sem, A sec** |
| **Github Repository:** | **Lepakshi-044** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **Structure:**   |  |  | | --- | --- | | |  | | --- | | A structure is a user-defined data type available in C that allows to combining data items of different kinds. Structures are used to represent a record. Defining a structure: To define a structure, you must use the struct statement. The struct statement defines a new data type, with more than or equal member. The format of the struct statement is as follows:  struct [structure name]  {  member definition;  member definition;  ...  member definition;  };  **Union:**  A union is a special data type available in C that allows storing different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple purposes. Defining a Union: To define a union, you must use the union statement in the same way as you did while defining a structure. The union statement defines a new data type with more than one member for your program. The format of the union statement is as follows:  union [union name]  {  member definition;  member definition;  ...  member definition;  };  Similarities between Structure and Union:  • Both are user-defined data types used to store data of different types as a single unit.  • Their members can be objects of any type, including other structures and unions or arrays. A member can also consist of a bit field.  • Both structures and unions support only assignment = and sizeof operators. The two structures or unions in the assignment must have the same members and member types.  • A structure or a union can be passed by value to functions and returned by value by functions. The argument must have the same type as the function parameter. A structure or union is passed by value just like a scalar variable as a corresponding parameter.  • ‘.’ operator is used for accessing members.  **Error Handling:**  Error handling features are not supported by C programming, which is known as exception handling in C++ or in other OOP (Object Oriented Programming) languages. However, there are  few methods and variables available in C's header file *error.h* that is used to locate errors using return values of the function call. In C, the function return *NULL* or *-1* value in case of any error, and there is a global variable *errno* which sets the error code/number. Hence, the return value can be used to check error while programming.  **File Handling in C:**  Different operations that can be performed on a file are:  1. Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w++”)  2. Opening an existing file (fopen)  3. Reading from file (fscanf or fgets)  4. Writing to a file (fprintf or fputs)  5. Moving to a specific location in a file (fseek, rewind)  6. Closing a file (fclose) | | |