**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **18-06-2020** | **Name:** | **Dhanya Shetty** |
| **Course:** | **C PROGRAMMING** | **USN:** | **4AL17EC026** |
| **Topic:** | **1.BASIC CONCEPTS**  **2.CONDITIONAL LOOPS**  **3.FUNCTIONS, ARRAYS &POINTERS**  **4.STRINGS & FUNCTION POINTERS** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Dhanya Shetty\_026** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| C:\Users\Hp\Desktop\report\18june1111.PNG  C:\Users\Hp\Desktop\june182222.PNG  **C:\Users\Hp\Desktop\18june3333.PNG**  **C Programming :**  **C** is highly portable and is **used for** scripting system applications which form a major part of Windows, UNIX, and Linux operating system. **C** is a general-purpose **programming language** and can efficiently work on enterprise applications, games, graphics, and applications requiring calculations, etc.  **Benefits of C language over other programming languages :**  [C](https://www.geeksforgeeks.org/c/) is a middle-level programming language developed by Dennis Ritchie during the early 1970s while working at AT&T Bell Labs in the USA. The objective of its development was in the context of the re-design of the UNIX operating system to enable it to be used on multiple computers.  Earlier the language B was now used for improving the UNIX system. Being a high-level language, B allowed much faster production of code than in assembly language. Still, B suffered from drawbacks as it did not understand data-types and did not provide the use of “structures”.  These drawbacks became the driving force for Ritchie for development of a new programming language called C. He kept most of language B’s syntax and added data-types and many other required changes. Eventually, C was developed during 1971-73, containing both high-level functionality and the detailed features required to program an operating system. Hence, many of the UNIX components including UNIX kernel itself were eventually rewritten in C.  **Basic concepts :**  It was mainly developed as a system **programming language** to write an operating system. The main features of **C language** include low-level access to memory, a simple set of keywords, and clean style, these features make **C language** suitable for system programming’s like an operating system or compiler development.  **What are keywords in C?**  In **C** programming, a **keyword** is a word that is reserved by a program because the word has a special meaning. **Keywords** can be commands or parameters. Every programming language has a set of **keywords** that cannot be used as variable names. **Keywords** are sometimes called reserved names .  **keywords in C :**  **Keywords** are part of the syntax and they cannot be used as an identifier. For **example**: int money; Here, int is a **keyword** that indicates money is a variable of type int (integer).  **Conditional Loops :**  In while **loop**, a **condition** is evaluated before processing a body of the **loop**. If a **condition** is true then and only then the body of a **loop** is executed. ... Once the **condition** becomes false, the control goes out of the **loop**. After exiting the **loop**, the control goes to the statements which are immediately after the **loop**.  **3 types of loops :**  **Loops** are control structures used to repeat a given section of code a certain number of times or until a particular condition is met. Visual Basic has **three** main **types of loops**: for.. next **loops**, do **loops** and while **loops**.  **What is loop in C and its types?**  C - Loops   |  |  | | --- | --- | | **Sr.No.** | **Loop Type & Description** | | 1 | while loop Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body. | | 2 | for loop Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. |   **Conditional statements in C :**  **Conditional statements** help you to make a decision based on certain conditions. These conditions are specified by a set of **conditional statements** having boolean expressions which are evaluated to a boolean value true or false. There are following types of **conditional statements in C**. If **statement**. If-Else **statement.**  **What is Loop and types?**  In computer science, a **loop** is a programming structure that repeats a sequence of instructions until a specific condition is met. Programmers use **loops** to cycle through values, add sums of numbers, repeat functions, and many other things. ... Two of the most common **types** of **loops** are the while **loop** and the for **loop**.  **Functions :**  **There are two types of functions in C programming:**   * Library **Functions**: are the **functions** which are declared in the **C** header files such as scanf(), printf(), gets(), puts(), ceil(), floor() etc. * User-defined **functions**: are the **functions** which are created by the **C programmer**, so that he/she can use it many times.   **There can be 4 different types of user-defined functions, they are:**   * Function with no arguments and no return value. * Function with no arguments and a return value. * Function with arguments and no return value. * Function with arguments and a return value.   **Functions** are used for Placing or Storing the Code which is to be Repeated Several Times. For **Example**, if we need Same Code, then we must have to Write that Code Again and Again So that for Removing this Task, we uses **functions**.  **Advantages of Function :**  Avoid repetition of codes. Increases program readability. Divide a complex problem into simpler ones. Reduces chances of error.  **Arrays :**  **Arrays** a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An **array** is used to store a collection of data.  **Whats is an array?**  An arrangement of objects, pictures, or numbers in columns and rows is called an **array**. **Arrays** are useful representations of multiplication concepts. This **array** has 4 rows and 3 columns. It can also be described as a 4 by 3 **array**. ... When equal groups are arranged in equal rows, an **array** is formed.  Computer **Programming** - **Arrays**. ... An **array** is a data structure, which can store a fixed-size collection of elements of the same data type. An **array** is used to store a collection of data, but it is often more useful to think of an **array** as a collection of variables of the same type.  **Pointer in C programming :**  A **pointer** is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a **pointer** before using it to store any variable address.  **What is pointer in C with example?**  A **pointer** is a variable that stores the address of another variable. Unlike other variables that hold values of a certain type, **pointer** holds the address of a variable. For **example**, an integer variable holds (or you can say stores) an integer value, however an integer **pointer** holds the address of a integer variable.  **List Of Pointers In C Programming :**   * Null Pointer. NULL Pointer is a pointer which is pointing to nothing. ... * Dangling Pointer * Generic Pointers * Wild Pointer * Complex Pointers * Near Pointer * Far Pointer * Huge Pointer   **Strings in C language?**  In **C programming**, a **string** is a sequence of characters terminated with a null character \0 . For example: char **c**[] = "**c string**"; When the compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character \0 at the end by default.  A **string** is a data type used in programming, such as an integer and floating point unit, but is used to represent text rather than numbers. It is comprised of a set of characters that can also contain spaces and numbers. For **example**, the word "hamburger" and the phrase "I ate 3 hamburgers" are both **strings**.  **string manipulation :**  **String manipulation** (or **string** handling) is the process of changing, parsing, splicing, pasting, or analysing **strings**. ... Typically, most programming languages provide a **string** data type that holds a sequence of characters.  **What is difference between character array and string in C?**  The **difference between** just an **array** of **characters** and a **string in C** is the addition of a NULL **character** (\0) at the end. ... **Strings** can be declared using the same method used to specify values **in a character array** or we can use a shortcut by including a **string** in double quotes.  **String Manipulation Techniques**   1. Extract or truncate the first few characters of a string, 2. Extract or truncate some characters at the end of the string, 3. Find out the length (number of characters) of a string, 4. Convert a string from lowercase to UPPERCASE or vice-versa, 5. Check if a character has been used in a string,   **Webinar on “BLOCKCHAIN TECHNOLOGY” conducted by EMURGO hosted by Mr. Raghu Raman on 17th June,2020-Wednesday**  C:\Users\Hp\Desktop\report\Screenshot_20200617-160703_Zoom.jpg  **C:\Users\Hp\Desktop\report\Screenshot_20200617-165938_Zoom.jpg** |
|  |

|  |
| --- |
|  |
|  |
|  |