**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **20-06-2020** | **Name:** | **Dhanya Shetty** |
| **Course:** | **C PROGRAMMING** | **USN:** | **4AL17EC026** |
| **Topic:** | **1.FILES & ERROR HANDLING**  **2.THE PREPROCESSOR** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Dhanya Shetty\_026** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| C:\Users\Hp\Desktop\report\20june111.PNG  C:\Users\Hp\Desktop\report\20june222.PNG  **C:\Users\Hp\Desktop\report\20june333.PNG**  **C:\Users\Hp\Desktop\report\20june444.PNG**  **C:\Users\Hp\Desktop\report\20june555.PNG**  **C:\Users\Hp\Desktop\report\20june11111.PNG**  **C:\Users\Hp\Desktop\report\20june222222.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.  **Files and Error Handling :**  The **C** programming language provides perror () and strerror () **functions** which can be used to display the text message associated with err no. The perror() **function** displays the string you pass to it, followed by a colon, a space, and then the textual representation of the current errno value.  **Functions of error handler :**  The tasks of the **Error Handling** process are to detect each **error**, report it to the user, and then make some recover strategy and implement them to handle **error**. During this whole process processing time of program should not be slow. An **Error** is the blank entries in the symbol table.  **The Preprocessor :**  In [computer science](https://en.wikipedia.org/wiki/Computer_science), a **pre-processor** is a [program](https://en.wikipedia.org/wiki/Computer_program) that processes its input data to produce output that is used as input to another program. The output is said to be a **pre-processed** form of the input data, which is often used by some subsequent programs like [compilers](https://en.wikipedia.org/wiki/Compiler). The amount and kind of processing done depends on the nature of the pre-processor; some pre-processors are only capable of performing relatively simple textual substitutions and [macro](https://en.wikipedia.org/wiki/Macro_(computer_science)) expansions, while others have the power of full-fledged [programming languages](https://en.wikipedia.org/wiki/Programming_language).  A common example from [computer programming](https://en.wikipedia.org/wiki/Computer_programming) is the processing performed on [source code](https://en.wikipedia.org/wiki/Source_code) before the next step of compilation. In some [computer languages](https://en.wikipedia.org/wiki/Computer_language) (e.g., [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [PL/I](https://en.wikipedia.org/wiki/PL/I_(programming_language))) there is a phase of [translation](https://en.wikipedia.org/wiki/Compiler) known as *pre-processing*. It can also include macro processing, file inclusion and language extensions.  The most common example of this is the [C preprocessor](https://en.wikipedia.org/wiki/C_preprocessor), which takes lines beginning with '#' as [directives](https://en.wikipedia.org/wiki/Directive_(programming)). Because it knows nothing about the underlying language, its use has been criticized and many of its features built directly into other languages. For example, macros replaced with aggressive in lining and templates, includes with compile-time imports (this requires the preservation of type information in the object code, making this feature impossible to retrofit into a language); [conditional compilation](https://en.wikipedia.org/wiki/Conditional_compilation) is effectively accomplished with if-then-else and dead code elimination in some languages. However, a key point to remember is that all preprocessor directives should start on a new line.  The most common use of the **preprocessor** is to include header files. In C and C++, all symbols must be declared in a file before they can used. They don't always **need** to be defined\*, but the compiler needs to know they exist somewhere. ... The most common use of the **preprocessor** is to include header files.  **C:\Users\Hp\Desktop\report\c programming certificate.PNG** |
|  |

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