**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **25/06/2020** | **Name:** | **PRIYA P RAO** |
| **Course:** | **C++ Programming** | **USN:** | **4AL18EC041** |
| **Topic:** | * **Inheritance and Polymorphism** * **Templates, Exceptions and Files** | **Semester & Section:** | **4th sem ‘A’ section.** |
| **Github Repository:** | **Priya-Rao** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **C:\Users\Pawan\Desktop\r1.PNG** |
| **In today’s session I have learnt about:**  **Chapter 1: Inheritance and Polymorphism**   * **Inheritance :**   **Inheritance is a process in which one object acquires all the properties and behaviors of its parent object automatically.**   * **Protected members :**   **A protected member variable or function is very similar to a private member but it provided one additional benefit that they can be accessed in child classes which are called derived classes.**   * **Derived class constructor and destructor :**   **Base class constructors are called first and the derived class constructors are called next in single inheritance. Destructor is called in reverse sequence of constructor invocation i.e. The destructor of the derived class is called first and the destructor of the base is called next.**   * **Polymorphism :**   **C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function.**   * **Virtual functions :**   **A C++ virtual function is a member function in the base class that you redefine in a derived class. It is declared using the virtual keyword.**   * **Abstract classes :**   **An abstract class in C++ is a class that has at least one pure virtual function (i.e., a function that has no definition). The classes inheriting the abstract class must provide a definition for the pure virtual function; otherwise, the subclass would become an abstract class itself.**  **Chapter 2: Templates, Exceptions and Files**   * **Function templates :**   **Function templates are special functions that can operate with generic types.**   * **Function templates with multiple parameters :**   **Line 1 declares a class template which has two template parameters. Okay, the first parameter is the type of the elements and the second parameter stands for the container. Let's have a closer look at the second parameter: template <typename, typename> class Cont >.**   * **Class templates :**   **A class template provides a specification for generating classes based on parameters. Class templates are generally used to implement containers. A class template is instantiated by passing a given set of types to it as template arguments.**   * **Template specialization :**   **The act of creating a new definition of a function, class, or member of a class from a template declaration and one or more template arguments is called template instantiation.**   * **Exceptions :**   **A C++ exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero. Exceptions provide a way to transfer control from one part of a program to another. C++ exception handling is built upon three keywords: try, catch, and throw.**   * **More on exceptions :**   **C++ Exception Handling**   * **throw − A program throws an exception when a problem shows up. ...** * **catch − A program catches an exception with an exception handler at the place in a program where you want to handle the problem. ...** * **try − A try block identifies a block of code for which particular exceptions will be activated.** * **Working with files :**   **C++ provides us with the following operations in File Handling:**   1. **Creating a file: open()** 2. **Reading data: read()** 3. **Writing new data: write()** 4. **Closing a file: close()**  * **More on files :**   **It is possible to open more than one file at a time. Simply declare and use a separate stream variable name (fout, fin, fout2, fin2 -- file pointer) for each file.** |