**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **26-06-2020** | **Name:** | **Dhanya Shetty** |
| **Course:** | **C PROGRAMMING** | **USN:** | **4AL17EC026** |
| **Topic:** | **1.CHALLENGE 1**  **2.CHALLENGE 2** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Dhanya Shetty\_026** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| C:\Users\Hp\Desktop\report\26june111.PNG  C:\Users\Hp\Desktop\report\26june222.PNG  **C:\Users\Hp\Desktop\report\26june333.PNG**  **C:\Users\Hp\Desktop\report\26june444.PNG**  **C:\Users\Hp\Desktop\report\26june555.PNG**  **C:\Users\Hp\Desktop\report\26june666.PNG**  **C++ Programming :**  **C++** is a powerful general-purpose **programming language**. It can be used to develop operating systems, browsers, games, and so on. **C++** supports different ways **of programming** like procedural, object-oriented, functional, and so on. This makes **C++** powerful as well as flexible.  C++ was designed with a bias toward [system programming](https://en.wikipedia.org/wiki/System_programming) and [embedded](https://en.wikipedia.org/wiki/Embedded_software), resource-constrained software and large systems, with [performance](https://en.wikipedia.org/wiki/Performance_(software)), efficiency, and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including [desktop applications](https://en.wikipedia.org/wiki/Application_software), [video games](https://en.wikipedia.org/wiki/Video_game_development), [servers](https://en.wikipedia.org/wiki/Server_(computing)) (e.g. [e-commerce](https://en.wikipedia.org/wiki/E-commerce), [Web search](https://en.wikipedia.org/wiki/Web_search_engine), or [SQL](https://en.wikipedia.org/wiki/SQL) servers), and performance-critical applications (e.g. [telephone switches](https://en.wikipedia.org/wiki/Telephone_switches) or [space probes](https://en.wikipedia.org/wiki/Space_probes)).  C++ is standardized by the [International Organization for Standardization](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO), with the latest standard version ratified and published by ISO in December 2017 as [*ISO/IEC 14882:2017*](https://en.wikipedia.org/wiki/C%2B%2B#Standardization) (informally known as [C++17](https://en.wikipedia.org/wiki/C%2B%2B17)). The C++ programming language was initially standardized in 1998 as *ISO/IEC 14882:1998*, which was then amended by the [C++03](https://en.wikipedia.org/wiki/C%2B%2B03), [C++11](https://en.wikipedia.org/wiki/C%2B%2B11) and [C++14](https://en.wikipedia.org/wiki/C%2B%2B14) standards. The current C++17 standard supersedes these with new features and an enlarged [standard library](https://en.wikipedia.org/wiki/C%2B%2B#Standard_library). Before the initial standardization in 1998, C++ was developed by Danish computer scientist [Bjarne Stroustrup](https://en.wikipedia.org/wiki/Bjarne_Stroustrup) at [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs) since 1979 as an extension of the [C language](https://en.wikipedia.org/wiki/C_(programming_language)); he wanted an efficient and flexible language similar to C that also provided [high-level features](https://en.wikipedia.org/wiki/High-level_programming_language) for program organization.  [C++20](https://en.wikipedia.org/wiki/C%2B%2B20) is the next planned standard, keeping with the current trend of a new version every three years.  **History :**  In 1979, [Bjarne Stroustrup](https://en.wikipedia.org/wiki/Bjarne_Stroustrup), a Danish [computer scientist](https://en.wikipedia.org/wiki/Computer_scientist), began work on "C with [Classes](https://en.wikipedia.org/wiki/Class_(computer_programming))", the predecessor to C++.The motivation for creating a new language originated from Stroustrup's experience in programming for his PhD thesis. Stroustrup found that [Simula](https://en.wikipedia.org/wiki/Simula) had features that were very helpful for large software development, but the language was too slow for practical use, while [BCPL](https://en.wikipedia.org/wiki/BCPL) was fast but too low-level to be suitable for large software development. When Stroustrup started working in [AT&T Bell Labs](https://en.wikipedia.org/wiki/AT%26T_Bell_Labs), he had the problem of analyzing the [UNIX](https://en.wikipedia.org/wiki/Unix) [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) with respect to [distributed computing](https://en.wikipedia.org/wiki/Distributed_computing). Remembering his Ph.D. experience, Stroustrup set out to enhance the [C](https://en.wikipedia.org/wiki/C_(programming_language)) language with [Simula](https://en.wikipedia.org/wiki/Simula)-like features. C was chosen because it was general-purpose, fast, portable and widely used. As well as C and Simula's influences, other languages also influenced this new language, including [ALGOL 68](https://en.wikipedia.org/wiki/ALGOL_68), [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language)), [CLU](https://en.wikipedia.org/wiki/CLU_(programming_language)) and [ML](https://en.wikipedia.org/wiki/ML_(programming_language)).  Initially, Stroustrup's "C with Classes" added features to the C compiler, Cpre, including [classes](https://en.wikipedia.org/wiki/Class_(computer_programming)), [derived classes](https://en.wikipedia.org/wiki/Derived_class), [strong typing](https://en.wikipedia.org/wiki/Strong_typing), [inlining](https://en.wikipedia.org/wiki/Inlining) and [default arguments](https://en.wikipedia.org/wiki/Default_argument).  In 1982, Stroustrup started to develop in C) after going through several other names. New features were added, including [virtual functions](https://en.wikipedia.org/wiki/Virtual_function), function name and [operator overloading](https://en.wikipedia.org/wiki/Operator_overloading), [references](https://en.wikipedia.org/wiki/Reference_(C%2B%2B)), constants, type-safe free-store memory allocation (new/delete), improved type checking, and BCPL style single-line comments with two forward slashes (*//*). Furthermore, Stroustrup developed a new, standalone compiler for C++, [Cfront](https://en.wikipedia.org/wiki/Cfront).  In 1985, the first edition of [*The C++ Programming Language*](https://en.wikipedia.org/wiki/The_C%2B%2B_Programming_Language) was released, which became the definitive reference for the language, as there was not yet an official standard. The first commercial implementation of C++ was released in October of the same year.  In 1989, C++ 2.0 was released, followed by the updated second edition of *The C++ Programming Language* in 1991. New features in 2.0 included multiple inheritance, abstract classes, static member functions, [const member functions](https://en.wikipedia.org/wiki/Const_correctness), and protected members. In 1990, *The Annotated C++ Reference Manual* was published. This work became the basis for the future standard. Later feature additions included [templates](https://en.wikipedia.org/wiki/Template_(programming)), [exceptions](https://en.wikipedia.org/wiki/Exception_handling), [namespaces](https://en.wikipedia.org/wiki/Namespaces), new [casts](https://en.wikipedia.org/wiki/Cast_(computer_science)), and a [Boolean type](https://en.wikipedia.org/wiki/Boolean_datatype).  C++ is a general purpose programming language and widely used now a days for competitive programming. It has imperative, object-oriented and generic programming features. C++ runs on lots of platform like Windows, Linux, Unix, Mac etc.  Among open source **software** one best is Eclipse it advance functionality for C/**C++ programmers**. It has many new features like syntax highlighting and auto code. It is compatible with all windows operating systems, Linux and Mac OS X. Java Run time environment is necessary to compile your C/**C++ Programs** on your computer.  C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.  Some of the ***features & key-points*** to note about the programming language are as follows:   * **Simple**: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich libray support and a variety of data-types. * **Machine Independent but Platform Dependent**: A C++ executable is not platform-independent (compiled programs on Linux won’t run on Windows), however they are machine independent. * **Mid-level language**: It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.) * **Rich library support**: Has a rich library support (Both standard ~ built-in data structures, algorithms etc.) as well 3rd party libraries (e.g. Boost libraries) for fast and rapid development. * **Speed of execution**: C++ programs excel in execution speed. Since, it is a compiled language, and also hugely procedural. Newer languages have extra in-built default features such as grabage-collection, dynamic typing etc. which slow the execution of the program overall. Since there is no additional processing overhead like this in C++, it is blazing fast. * **Pointer and direct Memory-Access**: C++ provides pointer support which aids users to directly manipulate storage address. This helps in doing low-level programming (where one might need to have explicit control on the storage of variables). * **Object-Oriented**: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications ca be built. Procedural code becomes difficult to maintain as code-size grows. * **Compiled Language**: C++ is a compiled language, contributing to its speed.   **Applications**  C++ finds varied usage in applications such as:   * Operating Systems & Systems Programming. E.g. *Linux-based OS (Ubuntu etc.)* * Browsers *(Chrome & Firefox)* * Graphics & Game engines *(Photoshop, Blender, Unreal-Engine)* * Database Engines *(MySQL, MongoDB, Radis etc.)* * Cloud/Distributed Systems   Here are some awesome facts about C++ that may interest you:   1. The name of C++ signifies the evolutionary nature of the changes from C. “++” is the C increment operator. 2. C++ is one of the predominant languages for the development of all kind of technical and commercial software. 3. C++ introduces Object-Oriented Programming, not present in C. Like other things, C++ supports the four primary features of OOP: encapsulation, polymorphism, abstraction, and inheritance. 4. C++ got the OOP features from Simula67 Programming language. 5. A function is a minimum requirement for a C++ program to run.(at least main() function)   **Inheritance**is one of the most important concepts of object-oriented programming. Inheritance allows us to define a class based on another class. This facilitates greater ease in creating and maintaining an application.  The class whose properties are inherited by another class is called the **Base** class. The class which inherits the properties is called the **Derived** class. For example, the **Daughter**class (derived) can be inherited from the **Mother**class (base). The derived class inherits all feature from the base class, and can have its own additional features.  C++ polymorphism means that a call to a member function will cause a **different**implementation to be executed depending on the **type**of object that invokes the function.  The previous example demonstrates the use of base class pointers to the derived classes. Why is that useful? Continuing on with our game example, we want every Enemy to have an **attack ()** function.  Functions and classes help to make programs easier to write, safer, and more maintainable. However, while functions and classes do have all of those advantages, in certain cases they can also be somewhat limited by C++'s requirement that you specify types for all of your parameters.  Template functions can save a lot of time, because they are written only once, and work with different types. Template functions reduce code maintenance, because duplicate code is reduced significantly.  **CERTIFICATE :**  **C:\Users\Hp\Desktop\report\26june111111.PNG** |

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