The C Programming Language

Background and Rationale

- Development of the C language began in 1972 at Bell Labs.
- C was designed by Dennis Ritchie as a successor to the B programming language designed by Ken Thompson.
- The development of C was closely tied to the development of the UNIX operating system, with UNIX eventually being made based on C.
- C was designed to provide low-level access to memory and efficiently map onto machine instructions.

Classification

- C is a general-purpose language, which means that it can be applied to a broad variety of circumstances.
- C follows the imperative programming paradigm, which means that its instructions directly change the state of the computer. This means the C programmer not only codes what they want to be done, but how they want it done.
- C also follows the procedural programming paradigm, which is a type of imperative programming in which the programmer designs software by using functions.
- C is a middle-level language, which means the compiler doesn’t do a lot of interpretation and the user gains more direct access to the computer’s resources than a high-level language like Python.
- C is a standardized language.

Evaluation

- C isn’t as readable as some higher level languages, but it maintains an impressive degree of readability while remaining close to bare-metal.
- C is less writable than other languages, but there are scenarios where it is by far the most productive choice. One such scenario is embedded systems programming where often the only viable alternative to C is assembly, which is far less writable.
- C isn’t very reliable as its particularly prone to errors like race conditions and memory leaks. This is partly inevitable since C is a relatively low-level language geared towards efficiency.
- C isn’t particularly costly since the tools needed to start coding, like GCC, are open source. But it may take a little longer to work in C due to it being a little less writable and readable.

Evaluation

- Unlike assembly, C is a portable language. This means that it isn’t tied to any hardware or system, so the developer can expect it to work on any computer.
- C excels in generality, which is what makes it such a widely applicable programming language.
- Due to its age and importance to the IT field, C has a well-established community which produces lots of textbooks and other learning material. It has a ton of industry backing too.
- C is used in a variety of very major projects. Microsoft and Apple base their operating systems on C. Linux, which is the operating system of choice for servers, is also based on C.

Code Sample

Works Cited

https://en.wikipedia.org/wiki/C_(programming_language)
https://www.section.io/engineering-education/history-of-c-programming-language/
https://en.wikipedia.org/wiki/General-purpose_programming_language
https://www.computerhope.com/jargon/i/imp-programming.htm
https://en.wikipedia.org/wiki/Imperative_programming
https://www.allaboutcircuits.com/technical-articles/assembly-vs-c-why-learn-assembly/
https://www.geeksforgeeks.org/why-learning-c-programming-is-a-must/
https://rust-class.org/0/pages/using-rust-for-an-undergraduate-os-course.html
https://en.wikipedia.org/wiki/GNU_Compiler_Collection
https://www.log2base2.com/C/basic/introduction-to-c-language.html
https://www.toptal.com/c/after-all-these-years-the-world-is-still-powered-by-c-programming