

Nombre: Miguel Alejandro Santiago Pérez

No. de Matrícula.: zap362_____

Materia: Fundamentos de Programacion

Grupo: Dev-1

Turno: Matutino

Carrera: Desarrollo de Software Interactivo y Videojuegos_____

Tema: Palabras Reservadas

No: T.3

Fecha propuesta: 22/03/2021

Fecha de Entrega: 29/03/2021

Escuela: Instituto Universitario Amerike

Plantel Zapopan

Calle: Calle Monetmorelos No: 3503

Colonia: Rinconada de la Calma

C.P.: 45080

Teléfono: 33363256100

Ciudad: Zapopan_____



Firma del alumno (a)

Firma de revisión fecha

Qué se evalúa:	10 pts.	7 pts.	4pts.	Pts.
Entrega electrónica	Es en tiempo y forma al iniciar la clase. (1 pts.)	Después de 30 minutos de iniciada la clase. (.7 pts.)	Al minuto 40. (Posteriormente ya no se reciben) (.4pts.)	
Del formato.	Cumple con todos los elementos solicitados. (1 pts.)	No cumple con dos elementos solicitados. (.7 pts.)	No cumple con tres o más elementos solicitados. (.4pts.)	
La ortografía.	Tiene dos errores ortográficos. (1 pts.)	Tiene de tres a cuatro errores ortográficos. (.7 pts.)	Tiene cinco o más errores ortográficos. (.4pts.)	
Del tema y objetivo.	La teoría y ejemplos corresponden al tema tratado. (1 pts.)	La teoría o ejemplos no corresponden al tema tratado. (.7 pts.)	La teoría y ejemplos no corresponden al tema tratado. (.4pts.)	
El programa y los cálculos.	Los parámetros y componentes corresponden al 100% de lo planeado. (1 pts.)	El programa arroja un error o componente no corresponden al 100% de lo planeado. (7 pts.)	El programa arroja dos errores o componentes no corresponden al 100% de lo calculado. (.4pts.)	
Diagramas.	Los diagramas a bloques, de flujo y esquemáticos son acorde al de la práctica y siguen una secuencia lógica. (1 pts.)	Los diagramas a bloques, o de flujo o esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.7 pts.)	Los diagramas a bloques, de flujo y esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.4pts.)	
La tabla de valores.	Los valores calculados y medidos presentan una desviación máxima del 10%. (1 pts.)	Los valores calculados y medidos presentan una desviación máxima del 15%. (.7 pts.)	Los valores calculados y medidos presentan una desviación máxima del 20%. (.4pts.)	
Las observaciones y conclusiones.	Son específicas y congruentes con la práctica. (1 pts.)	Las observaciones o conclusiones son específicas y congruentes con la práctica. (.7 pts.)	Las observaciones y las conclusiones no son específicas y congruentes con la práctica. (.4pts.)	
Bibliografía.	Es acorde al (los) tema (s) tratado (s) y está completa (1 pts.)	Es acorde a algún (os) tema (s) tratado (s), le falta algún elemento que la conforman (.7 pts.)	No es acorde al (los) tema (s) tratado (s), le faltan 2 elementos que la conforma (.4pts.)	
Fuentes de consulta.	Es acorde al (los) tema (s) tratado (s) (1 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.7 pts.)	Es acorde a algún (los) tema (s) tratado (s) (.4pts.)	

Índice El objetivo será investigar como C++ almacena memoria al declarar variables

Teoría “Los punteros (o apuntadores) son variables que se utilizan para almacenar direcciones de memoria, puntualmente las direcciones de memoria que fueron asignadas a variables convencionales en las que se almacenan datos de distinto tipo. Vale la pena entonces recordar que a todas las variables en C++ se les asigna un espacio de memoria en el cual se va almacenar el valor que se le asigne en algún punto de la aplicación a esa variable, el tamaño de dicho espacio va depender del tipo de dato que se pretende almacenar en la variable, del compilador y de la arquitectura del procesador. Cada uno de los espacios de memoria cuenta con una dirección para identificarlo, esta dirección es por lo general un número en representación hexadecimal. Es precisamente ese número correspondiente a la dirección lo que se almacena en un puntero.” [_ \(Game, s.f.\)](#)

“Las variables en C pueden ser de varios tipos y serán utilizadas en función del tipo de datos que queramos almacenar en ellas. Las variables NOMBRE, nombre, Nombre son tres variables totalmente distintas. (Lenguaje Case Sensitive) y el nombre de una variable no puede comenzar por número (pero puede contener varios) ni tener caracteres especiales (se admite el guión bajo).

Por ejemplo: numero1, j10a, num_alumno, serían variables válidas y 1abc, numero?, num/alumnos serían variables inválidas.

Según dónde estén declaradas, las variables pueden ser globales (declaradas fuera de todo procedimiento o función) o locales (declaradas dentro de un procedimiento o función). Las primeras serán accesibles desde todo el código fuente y las segundas sólo en la función donde estén definidas.

Además, toda variable puede cambiar su valor durante la ejecución del programa y ser desplegada las veces que creamos oportunas.

Tras estas pequeñas aclaraciones, seguiremos comentando el uso de variables con el tipo float. Este tipo sirve para almacenar números decimales o reales, así como el double. El rango de valores admitidos en tan amplio que rara vez se nos quedarán obsoletos.

El tipo char, capaz de almacenar un único carácter. Internamente, un carácter es almacenado como un número.” [_ \(Omijal, 2016 \)](#)

Observaciones C++ es capaz de darte la opción de decidir en donde se guardara la información e incluso direccionarla a otros lados

Conclusiones Ya sea con variables o punteros hay una amplia variedad de opciones para decidir de que forma se guardaran nuestros datos en C++

Bibliografía [C Con Clase | Curso C++ \(cap25\)](#)

Fuentes de consulta [Tutorial c++ \(omijal.org\)](#)

[Punteros en C++ - Manejo dinámico de memoria y Polimorfismo \(Práctica 4\) \(codingame.com\)](#)