

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.2

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 de esta investigación es conocer las palabras reservadas en el lenguaje C++ para programación

Teoría

1. **Auto:** “En CPP o C++ existe la palabra clave **“auto”** al declarar variables, su uso permite al programador ahorrar la declaración del tipo en una variable. Dicho de otra manera, **“auto”** permite la inicialización de variables sin especificar el tipo.” [\(Parzibyte, 2019\)](#)
2. **Const:** Esta variable impide que el texto o dato dentro una variable, grupo o clase pueda ser modificado por el usuario, en caso de ser modificado causaría un error de compilación
3. **Double:** “Una variable de tipo **“double”** puede almacenar valores decimales. Por ejemplo: **double** resultado; resultado = 3 / 2; cout << resultado << endl; Mostrará por pantalla **“1.5”**.” [\(jilella, 2019\)](#)
4. **Float:** “Las variables de este tipo almacenan números en formato de coma flotante, mantisa y exponente, para entendernos, son números con decimales. Son aptos para variables de tipo real, como por ejemplo el cambio entre euros y pesetas. O para números muy grandes, como la producción mundial de trigo, contada en granos.” [_\(tigrex, 2011\)](#)
5. **Int:** “Las variables enteras almacenan números enteros dentro de los límites de su tamaño, a su vez, ese tamaño depende de la plataforma del compilador, y del número de bits que use por palabra de memoria: 8, 16, 32... No hay reglas fijas para saber el mayor número que podemos almacenar en cada tipo: int, long int o short int; depende en gran medida del compilador y del ordenador.” [_\(tigrex, 2011\)](#)
6. **Short:** Puede ser utilizada para declarar un int del tamaño que busquemos, ya sea un short o un long
7. **Struct:** “Una estructura es un grupo de variables las cuales pueden ser de diferentes tipos sostenidas o mantenidas juntas en una sola unidad. La unidad es la estructura” [_\(Velázquez, 2010\)](#)
8. **Unsigned:** “Le indica a la variable que no va a llevar signo (sin valores negativos).” [_\(Delgado, 2020\)](#)
9. **Break:** “Significa detener la ejecución de un bucle y salirse de él.” [_\(Alvarez, 2001\)](#)
10. **Continue:** “Sirve para detener la iteración actual y volver al principio del bucle para realizar otra iteración, si corresponde.” [_\(Alvarez, 2001\)](#)
11. **Else:** Se puede definir como otro camino o ruta a seguir en caso de que la variable if no sea verdadera
12. **For:** “Un ciclo for es una estructura de control iterativa, que nos permite ejecutar de manera repetitiva un bloque de instrucciones, conociendo previamente un valor de inicio, un tamaño de paso y un valor final para el ciclo.” [_\(Gonzalez, 2020\)](#)
13. **Do:** “Los ciclos do-while son una *estructura de control cíclica*, los cuales nos permiten ejecutar una o varias líneas de código de forma repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final, hasta aquí son similares a los ciclos while, sin embargo el ciclo do-while nos permite añadir cierta ventaja adicional y esta consiste que nos da la posibilidad de ejecutar primero el bloque de instrucciones antes de evaluar la condición necesaria, de este modo los ciclos do-while, son más efectivos para algunas situaciones específicas.” [_\(Gonzalez, 2020\)](#)
14. **Extern:** Es usada para declarar que los datos tienen links externos.
15. **If:** “La sentencia **if** se le conoce como estructura de selección simple y su función es realizar o no una determinada acción o sentencia, basándose en el resultado de la evaluación de una expresión (verdadero o falso), en caso de ser verdadero se ejecuta la sentencia.” [_\(Omijal, 2016\)](#)
16. **Return:** “La sentencia **return** finaliza la ejecución de la función y especifica un valor para ser devuelto a quien llama a la función.” [_\(Contributors, 2021\)](#)

17. Static: “Al interior de la definición de una clase se utiliza la palabra reservada `static` para declarar miembros de la clase que no están ligados a ninguna instancia (objeto) de la misma.” [_\(Game, s.f.\)](#)
18. Unión: “La definición de "unión" es similar a la de "estructura", La diferencia entre las dos es que en una estructura, los miembros ocupan diferentes áreas de la memoria, pero en una unión, los miembros ocupan la misma área de memoria.” [_\(WikiBooks, 2018\)](#)
19. While: “Los ciclos while son también una estructura cíclica, que nos permite ejecutar una o varias líneas de código de manera repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final que esperamos, los ciclos while, no dependen directamente de valores numéricos, sino de valores booleanos, es decir su ejecución depende del valor de verdad de una condición dada, verdadera o falso, nada más.” [_\(Gonzalez, Programar Ya, 2018\)](#)
20. Long: “El formato de variable numérica de tipo extendido “long” se refiere a números enteros (tipo 32 bits = 4 bytes) sin decimales que se encuentran dentro del rango -2147483648 a 2147483647.” (Bañales, 2017)
21. Signed: “Le indica a la variable que va a llevar signo.” [_\(Delgado, 2020\)](#)
22. Switch: “Los condicionales Switch, son una estructura de control condicional, que permite definir múltiples casos que puede llegar a cumplir una variable cualquiera, y qué acción tomar en cualquiera de estas situaciones, incluso es posible determinar qué acción llevar a cabo en caso de no cumplir ninguna de las condiciones dadas.” [_\(Gonzalez, Programar Ya, 2021\)](#)
23. Void: Es usada cuando la variable no devuelve un dato pero es señalada como un puntero
24. Case: Es utilizado para que la toma de decisiones múltiples sea mas rápida
25. Default: es opcional, en situaciones en que la variable no corresponda con ninguna de las expresadas.
26. Enum: Define una enumeración
27. Go to: “La instrucción de salto goto se puede usar en un programa, para transferir incondicionalmente el control del mismo a la primera instrucción después de una etiqueta, o dicho de otra forma, al ejecutar una instrucción goto, el control del programa se transfiere (salta) a la primera instrucción después de una etiqueta. Una etiqueta se define mediante su nombre (identificador) seguido del carácter dos puntos (:).” [_\(CarlosPes, s.f.\)](#)
28. Register: Indica al compilador una preferencia para que la variable se almacene en un registro de la CPU, si es posible, con el fin de optimizar su acceso y reducir el código. [_\(CCIA, s.f.\)](#)
29. Sizeof: Es el encargado de proporcionar la cantidad de almacenamiento para almacenar un objeto
30. Typedef: “es una palabra reservada en el lenguaje de programación C y C++. Su función es asignar un nombre alternativo a tipos existentes, a menudo cuando su declaración normal es aparatosa, potencialmente confusa o probablemente variable de una implementación a otra.” [_\(Wikipedia, 2020\)](#)
31. Volatile: “El modificador **volatile** quiere decir: Esta variable no sólo modifica tu programa sino alguien más.” [_\(Informaticos, 2018\)](#)
32. Char: “El tipo de datos char sirve para representar o contener un solo carácter. Un *byte* y un *char* son la misma cosa, excepto porque *char* se utiliza como carácter, o como parte de una cadena de caracteres, y no se puede utilizar en una expresión aritmética, mientras que un *byte* solamente puede ser usado como un tipo numérico.” [_\(Pascal, 2020\)](#)

Observaciones

Conclusiones

Bibliografía Manual de Programación en Lenguaje C++ [_\(Flores, 2008\)](#)