

Laboratorio 2

Estructuras de Control Condicional: Si y Selección

Condicional (SI)

La estructura de control más común es el condicional simple. Dicha estructura evalúa una expresión lógica, comúnmente llamada condición, para determinar si un bloque de instrucciones ha de ser ejecutado. En su forma más simple, únicamente se indican las instrucciones a ejecutar en caso de que la condición tenga valor Verdadero, mientras que la otra forma permite indicar dos bloques de instrucciones, uno para ser ejecutado cuando la condición evaluada resulta Verdadero y otro bloque de instrucciones para cuando la condición resulta Falso.

Pseudocódigo	C++
<u>Si</u> <condición> <u>entonces</u> [Secuencia de instrucciones]; <u>Fsi</u> ;	if (<condición>) { [Secuencia de instrucciones] }
<u>Si</u> <condición> <u>entonces</u> [Secuencia de instrucciones]; <u>Sino</u> [Secuencia de instrucciones]; <u>Fsi</u> ;	if (<condición>) { [Secuencia de instrucciones] } else { [Secuencia de instrucciones] }

NOTA: El uso de las llaves para delimitar un bloque de instrucciones es opcional en las estructuras de control en C++ siempre y cuando el bloque contenga una sola instrucción.

El anidamiento de estructuras de control condicional es posible de la misma forma que se realiza al utilizar pseudocódigo.

Pseudocódigo	C++
<u>Si</u> <condición1> <u>entonces</u> <u>Si</u> <condición2> <u>entonces</u> [Secuencia de instrucciones]; <u>Sino</u> [Secuencia de instrucciones]; <u>Fsi</u> ; <u>Fsi</u> ;	if (<condición1>) { if (<condición2>) { [Secuencia de instrucciones] } else { [Secuencia de instrucciones] } }

Selección múltiple

Como se vio en la sección anterior, la selección simple tiene traducción directa a C++. Sin embargo, la selección múltiple no solo no se puede traducir de forma directa sino que en muchos casos es simplemente imposible. A continuación se muestran lado a lado las definiciones léxicas de la selección múltiple en ambos lenguajes, de esta forma se intenta ilustrar las diferencias.

Pseudocódigo	C++
<u>Selección</u> <condición1>: [Secuencia de instrucciones 1]; <condición2>: [Secuencia de instrucciones 2]; ... <condiciónN>: [Secuencia de instrucciones N]; <u>Fselección;</u>	<pre>switch (<expresión>) { case <constante1>: [Secuencia de instrucciones 1]; break; case <constante2>: [Secuencia de instrucciones 2]; break; ... case <constanteN>: [Secuencia de instrucciones N]; break; default: [Secuencia de instrucciones]; }</pre>

En primer lugar se puede observar que la versión en pseudocódigo selecciona una sección de código asociada a una condición, mientras que la versión de C++ asocia la selección completa a una expresión y luego utiliza valores constantes para determinar el bloque de código a ejecutar. Esto significa que solo un subconjunto de las situaciones que pueden ser expresadas en la selección múltiple de pseudocódigo se puede traducir como selección múltiple a C++. En particular cuando todas las condiciones <condición1>, <condición2>, ..., <condiciónN> son de la forma <expresión>==<constante>. Es decir, si existe una expresión <expresión> para la cual si podemos reescribir la selección múltiple de la siguiente forma:

Pseudocódigo	C++
<u>Selección</u> <expresión>==<condición1>: [Secuencia de instrucciones 1]; <expresión>==<condición2>: [Secuencia de instrucciones 2]; ... <expresión>==<condiciónN>: [Secuencia de instrucciones N]; <u>Fselección;</u>	<pre>switch (<expresión>) { case <constante1>: [Secuencia de instrucciones 1]; break; case <constante2>: [Secuencia de instrucciones 2]; break; ... case <constanteN>: [Secuencia de instrucciones N]; break; }</pre>

Además, resalta la palabra clave `default` en la primera versión de C++, ésta cláusula especial es opcional y permite asociar una secuencia de instrucciones al caso en el que la expresión no se evalúe a ninguna de las constantes provistas. Es decir, si se tiene una expresión más en la versión de pseudocódigo de la forma “<expresión> ≠ <condición 1> y <expresión> ≠ <condición 2> y ... <expresión> ≠ <condición N>”, entonces se puede reemplazar ésta última por la cláusula `default`.

Pseudocódigo	C++
Selección <expresion> = <condicion1>: [Secuencia de instrucciones] <expresion> = <condicion 2>: [Secuencia de instrucciones] . . . <expresion> = <condicion N>: [Secuencia de instrucciones] <expresion> ≠ <condicion 1> y <expresion> ≠ <condicion 2> y ... <expresion> ≠ <condicion N>: [Secuencia de instrucciones] Fselección	switch (<expresión>) { case <constante1>: [Secuencia de instrucciones]; break; case <constante2>: [Secuencia de instrucciones]; break; . . . case <constanteN>: [Secuencia de instrucciones]; break; default: [Secuencia de instrucciones]; }

Finalmente, es importante mencionar que la expresión de la selección ha de ser de un tipo elemental de C++, es decir, de un tipo elemental en nuestro pseudocódigo que no sea String.

A continuación se muestra un ejemplo sencillo para la selección de una opción en un menú.

Pseudocódigo	C++
<u>Entero</u> entrada; <u>Escribir</u> ("Por favor suministre un número entero entre 0 y 2"); <u>Leer</u> (entrada); Selección entrada == 0: <u>Escribir</u> ("Seleccionó la opción 0"); entrada == 1: <u>Escribir</u> ("Seleccionó la opción 1"); entrada == 2: <u>Escribir</u> ("Seleccionó la opción 2");	int entrada; cout << "Por favor suministre un número entero entre 0 y 2" <<endl; cin >> entrada; switch (entrada) { case 0: cout << "Seleccionó la opción 0" <<endl; break; case 1: cout << "Seleccionó la opción 1" <<endl; break;

<pre> entrada < 0 o entrada > 2: Escribir("Opción inválida, fuera del rango indicado"); Fselección; </pre>	<pre> case 2: cout << "Seleccionó la opción 1" << endl; break; default: cout << "Opción inválida, fuera del rango indicado" << endl; break; } </pre>
--	---

Si la selección múltiple que deseamos escribir cumple con las condiciones descritas anteriormente, es posible realizar la traducción a la selección múltiple de C++. Sin embargo, cuando esto no sea posible se puede utilizar una secuencia de selecciones simples anidadas. Por ejemplo:

Pseudocódigo	C++
<pre> <u>Selección</u> <condición1>: [Secuencia de instrucciones 1]; <condición2>: [Secuencia de instrucciones 2]; ... <condiciónN>: [Secuencia de instrucciones N]; <u>Fselección;</u> </pre>	<pre> if (<condicion1>) { [Secuencia de instrucciones 1] } else { if (<condicion2>){ [Secuencia de instrucciones 2]; } else { if (<condicion3>) { [Secuencia de instrucciones 3]; } else { ... if (<condicionN>) { [Secuencia de instrucciones N]; } } } } } </pre>

EJERCICIOS A PROGRAMAR LABORATORIO 2

En este laboratorio debes desarrollar programas capaces usar la lectura y escritura estándar, declaración de variables, acciones elementales y prioridad de operadores, utilizando lenguaje C++ y creando programas que compilen y corran en el entorno Linux.

Ejercicios: todos deben agregarse en el mismo archivo .cpp, el cual debe compilar

NOTA: en todos los ejercicios debe solicitarse los valores de entrada al usuario, leerlos y realizar las validaciones necesarias en los datos de entrada.

1. Dados cuatro (4) números enteros positivos determine cuántos son iguales al mayor valor de ellos utilizando la estructura condicional Si simple.

Ejemplo, para los valores 12 50 -100 20 50 el mensaje a mostrar es:

El mayor valor es 50, la cantidad de valores iguales a 50 son 2

2. Cree un algoritmo que tome por entrada un número entero S que representa una cantidad en segundos, utilizando una variable de tipo int de C++ y operaciones equivalentes al div y mod. Luego de validar que el valor sea mayor o igual a cero, calcular la su equivalente a horas, minutos y segundos según las siguientes condiciones:

- Si el valor de S es menor a 3.600, indicar la cantidad de minutos y segundos a que es equivalente
- Si el valor es mayor o igual a 3.600 pero menor a 86.400, indicar la cantidad de horas, minutos y segundos a que es equivalente
- Si el valor es mayor a 86.400 pero menor o igual a 1.296.000, indicar la cantidad de días, horas, minutos y segundos a que es equivalente
- Para valores mayores a 1.296.000 se informa al usuario que no se procesa la cantidad por estar encima del límite superior del ejercicio

Mostrar el resultado del ejercicio según sea el caso mediante un mensaje equivalente a:

- *La cantidad de segundos es <valor_de_S>, equivalente a <cantidad_días> días, <cantidad_horas> horas, <cantidad_minutos> minutos y <cantidad_segundos>.*
- *La cantidad de segundos es <valor_de_S>, supera el límite superior del ejercicio (1296000)*

Ejemplo 1, para un valor de S = 45100 el mensaje de respuesta sería:

La cantidad de segundos es 45100, equivalente a 12 horas, 31 minutos y 40 segundos

Ejemplo 2, para un valor de S = 450120 el mensaje de respuesta sería:

La cantidad de segundos es 450120, equivalente a 5 días, 5 horas, 2 minutos y 0 segundos

Ejemplo 3, para un valor de S = 3600 el mensaje de respuesta sería:

La cantidad de segundos es 3600, equivalente a 1 hora, 0 minutos y 0 segundos

Ejemplo 4, para un valor de S = 1800000 el mensaje de respuesta sería:

La cantidad de segundos es 1800000, supera el límite superior del ejercicio (1296000)

3. Utilizando una estructura condicional de C++ equivalente a la **Selección múltiple**, realice un algoritmo que solicite al usuario un valor de tipo real x que representa el radio de una circunferencia o el valor del lado de un cuadrado, y solicite un valor entero op que representa una opción, a fin de calcular:
- Si op es 1: la longitud de la circunferencia
 - Si op es 2: el área de la circunferencia
 - Si op es 3: el perímetro del cuadrado
 - Si op es 4: el área del cuadrado
 - Si op no está en el rango 1 a 4: indicar que el valor de la opción suministrada está fuera de rango, no se realiza ningún cálculo.

Detalles de entrada

Se utilizará lectura estándar para obtener los datos de entrada suministrados por el usuario para cada ejercicio.

Detalles de la salida

Se escribirá por salida estándar una cadena o *string* que representa la respuesta a cada ejercicio.

Entrega del Ejercicio: Fecha de entrega: **hasta el viernes 19.diciembre.2014, 11:30 pm.**

Es OBLIGATORIO entregar la solución del laboratorio subiéndola a la plataforma Moodle en la cuenta de cada alumno.

En caso de problemas con la plataforma Moodle se debe enviar el archivo del laboratorio al correo electrónico del preparador de su grupo de laboratorio, con copia al correo del profesor de su sección y copia a la Profa. Yusneyi Carballo Barrera (yusneyi.carballo@ciens.ucv.ve) **antes de la hora tope de entrega, de lo contrario NO será corregido.**

Los correos electrónicos del grupo docente están en el sitio web de la materia, así como el correo electrónico del preparador de cada grupo de laboratorio.

Formato para el nombre del archivo, sin usar acentos o ñ:

<Seccion>Lab2<NombreyApellido>.cpp

Ejemplo, para el alumno con nombre José Antonio Mariñas

C1Lab2JoseMarinas.cpp

Restricciones

- El laboratorio debe ser realizarse en C++ de forma tal que **COMPILE** en el sistema operativo Ubuntu. Laboratorio que NO compila, NO se corrige.
- Se permite el uso de elementos de la Librería (o Biblioteca) Estándar de C++ (*stdio*, *iostream*, *string*, *cmath*, *math*, entre otras).

- Se debe usar lectura y escritura **estándar**.
- El ejercicio es individual y la entrega del mismo será **obligatoriamente** a través de la plataforma Moodle, también debe responderse el cuestionario que será habilitado.
- Cualquier copia será penalizada con CERO (0) puntos para todos los involucrados.

Material de Apoyo

- En el sitio web de la materia hay enlaces a sitios con información sobre programación con C++ (<http://www.ciens.ucv.ve/algoritmosyprogramacion/materiales.html>).
- En la Biblioteca Alonso Gamero y en la Bolsa del Libro están disponibles para préstamo muchos de los libros y Manuales de Programación con C++ indicados en la sección de Bibliografía recomendada al final de la página Materiales del sitio web.
- En el sitio web <https://ideone.com/> puede copiarse el código del programa, seleccionarse el lenguaje de programación, compilarse y probarse en línea.
- En <http://foro.noticias3d.com/vbulletin/showthread.php?t=400663> se indica cómo utilizar acentos en C++.

Consultas

- En el sitio web de la materia, sección Materiales, se indican los horarios en que los Preparadores estarán disponibles en la Sala de Micros para consultas de los laboratorios y proyectos.

GDAyP, 05 de Diciembre de 2014

Cuestionario Laboratorio 2

Estructuras de Control Condicional: Si y Selección

1. ¿Bajo qué condiciones es opcional el uso de llaves para delimitar una estructura condicional (*if*) en C++?
 - Cuando tenemos múltiples instrucciones
 - Cuando el valor de la expresión a evaluar es falso
 - Cuando el valor de la expresión a evaluar es verdadero
 - Siempre y cuando el bloque de instrucciones contenga una sola instrucción
2. ¿Es posible traducir la estructura de control condicional Selección de Pseudocódigo a lenguaje C++ (*switch ... case*) de forma directa para todos los casos?
 - Verdadero
 - Falso
3. ¿La selección en C++ (*switch ... case*) permite trabajar sólo con tipos de datos básicos del lenguaje, por ejemplo "string"?
 - Verdadero
 - Falso
4. Las expresiones lógicas pueden retornar los siguientes valores:
 - Cualquier número entero
 - Los valores verdadero y falso
 - Los caracteres 'V' o 'F'
 - Todas los valores anteriores

EN CASO DE NO TENER ACCESO AL MOODLE:

Aquellos alumnos que no tienen acceso al Moodle deben responder a las preguntas del cuestionario enviándolas por correo electrónico al preparador encargado de corregir su laboratorio, con el siguiente nombre de archivo:

<Seccion>Lab2<NombreyApellido>_Cuestionario.pdf (.doc)

ejemplo:

C3Lab2LuisGarcia_Cuestionario.pdf