



Universidad Católica
San Pablo

Ciencia de la Computación

Ciencia de la Computación I

Docente Alvaro Henry Mamani Aliaga

Capítulo 2 - Ejercicios Teóricos

Entregado el 27/08/2024

Rodrigo Farid Díaz Cucalón

Semestre II

2024-2

"El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo"

Ciencia de la Computación I - Introduction to C++ Programming

Chapter 2: Self Review Exercises

2.1. Revisa los espacios en blanco para los siguientes enunciados:

- a) Cada programa de C++ inicia su ejecución en la función main()

Según la página 87, cada programa debe contener una función y debe ser main, allí comienza la ejecución.

- b) Las líneas que comienzan con "#" son procesadas por el preprocesador antes de que el programa se compile.

La línea "#include <iostream>", incluida en la mayoría de programas es una directiva de preprocesamiento. El preprocesador indica al programa que use la información del archivo "<iostream>" antes de compilar.

- c) // indica que el resto de la línea es un comentario

Adicionalmente, se usa '/*' y '*/' indicando un comentario, un comentario en texto que el compilador ignore y no impacte en la ejecución.

- d) La igualdad se representa con: '==' y el operador de asignación es: '='

Para asignar un valor a una variable, previamente declarada (o en su defecto, en el momento que es declarada) se usa solo '=', para comparar dos variables bool o int se usa '=='.

- e) El símbolo "{" representa el inicio de la función main

La sintaxis de C++ requiere que cada función, ciclo o condicional contenga sus líneas entre corchetes "{}".

2.2 Indica si los siguientes enunciados son verdaderos o falsos, de ser falsos, explica el porque:

- a) Los comentarios causan que la computadora realice una acción cuando se ejecuta un programa y no son ignorados por C++

Falso: Todo tipo de comentarios son ignorados, su función es darle claridad al código y ayudar al programador a entender cierto código.

- b) Cuando un backslash "\" se utiliza en una cadena de caracteres en combinación con un carácter se forma una secuencia de escape

Verdadero: Los caracteres que siguen a un backslash son secuencias de escape y le indican al programa que un carácter especial debe ser mostrado.

- c) Los programas en C++ inician su ejecución desde la primera función encontrada

Falso: Aunque suele ser la primera, incluso de no ser el caso la ejecución empieza en `main()`

- d) Todas las variables deben ser especificadas con un tipo de variable

Verdadero: A diferencia de python, la sintaxis de C++ requiere que se especifique el tipo de dato para cada variable.

- e) Exactamente una función debe llamarse `main`

Verdadero: La función `main` es necesaria para cada programa.

- f) Los caracteres de una cadena aparecen tal como aparecen entre las dobles comillas

Verdadero

g) El enunciado 'return' es usado al final de 'main()'.

Verdadero: Aunque algunos programas admitan su exclusión, al ser un tipo de función 'int' espera que retorne una variable de tipo 'int' a través de 'return'.

h) Los operadores aritméticos $*$, $/$, $\%$, $+$ y $-$ tienen todos el mismo nivel de precedencia.

Falso: Los niveles de precedencia son:

Paréntesis ()	1°
Multiplicación *	2°
División /	
Residuo (Modulo) %	
Adición +	3°
Restación -	

i) Syntax Errors (errores sintácticos) también son llamados errores de compilación, errores al momento de compilar o errores de compilador y que son detectados durante la fase de compilación.

Verdadero.

2.3 Escribe un enunciado en C++ para cumplir los siguientes problemas. Asume que no se han usado las declaraciones "using".

a) Declara las variables: c, thisIsAVariable, q76354 y number, de tipo int y de valor 0.

```
int c{0}, thisIsAVariable{0}, q76354{0}, number{0};
```

b) Pide al usuario un número, seguido dos puntos (:) y un espacio para mover el cursor.

```
std::cout << "Ingresa un número: ";
```

c) Obtén el valor de tipo entero, dado por el usuario e imprimelo tras añadir 10;

`std::cin >> n; std::cout << n + 10;`

- d) Si la variable `value` es mayor a 15, restarle 5 de valor e imprimir el nuevo valor

`if (value > 15) value = value - 5; std::cout << value;`

- e) Imprime "This is a C++ program" en una línea

`std::cout << "This is a C++ program\n";`

- f) Imprime "This is a C++ program" en dos líneas, la primera terminando en "C++".

`std::cout << "This is a C++\nprogram\n";`

- g) Imprime "This is a C++ program" con cada palabra en una sola línea

`std::cout << "This is a C++\nprogram\n";` X
`std::cout << "This\nis\na\nC++\nprogram\n";` ✓

- h) Imprime el mensaje "This is a C++ program", cada palabra separada por una tabulación horizontal

`std::cout << "This\tis\t a\tC++\t program\n";`

- a) "int" le asigna el tipo entero a cada variable y el valor entre corchetes es el que se asigna
b) `std::cout` imprime los valores entre las dobles comillas
c) `std::cin` obtiene un valor para `n` y `cout` muestra el resultado de la suma

- d) if condiciona que `value` sea mayor que 15, de no el caso resta 5, al final se muestra su valor.

Para e), f), g) y h) se usan las secuencias de escape `\n` "new line" y `\t` para la tabulación en combinación con `std::cout` para formatear la línea según se pide

2.4 Escribe un enunciado (o comentario) que cumpla los siguientes ejercicios (asume que declaraciones "using" han sido usadas para cin, cout y endl)

a) Documenta que un programa calcula el producto de tres enteros:

```
// El programa multiplica 3 enteros y muestra el resultado
```

La documentación de un programa sirve para dar información sobre un programa y en este caso usamos un comentario

b) Declara las variables x, y, z y resultado de tipo int e inicializa cada una en '0'.

```
int x{0};  
int y{0};  
int z{0};  
int result{0};
```

c) Pide al usuario que ingrese 3 enteros.

```
cout << "Ingresa 3 enteros: ";
```

d) Lee tres enteros del teclado y almacena en las variables x, y y z.

```
cin >> x >> y >> z;
```

e) Calcula el producto de los tres enteros asignados a x, y y z y asigna el resultado en result.

```
result = x * y * z;
```

f) Imprime "El producto es " seguido por el valor de la variable result

```
cout << "El producto es: " << result << endl;
```

g) Retorna un valor desde main indicando que el programa termino.


```
return 0; }
```

2.6 Identifica y corrige los errores en cada uno de los siguientes enunciados (asumiendo que es el enunciado `std::cout` es usado).

a) `if (c < 7)`
`cout << "c es menor que 7 \n";`

En la línea "`if (c < 7)`" no está cerrado el paréntesis.

b) `if (c == 7) // != X, != ✓`
`cout << "c es igual o mayor que 7 \n";`

En la línea "`if (c == 7)`" el operador "`==`" no es correcto, el correcto sería "`!=`". Además, el programa en la condición no revisa si es igual o mayor, solo si es distinto.

Chapter 2: Exercises:

2.7 Diferente el significado de los objetos:

a) `#include <iostream>`:

Similar a la librería de python, `<iostream>` es un archivo de información parte de C++ que incluye funciones de input y output.

b) Operador (`<<`) de inserción:

El "left-shift operator" dirige los datos hacia la corriente de salida. El ingreso y salida se manejan usando estas corrientes de datos y la corriente de salida estándar está conectada a la pantalla.

2.8 Llena los espacios para cada uno de los enunciados.

a) Los comentarios se usan para documentar un programa y mejorar su legibilidad.

Al ser ignorados para la compilación y ejecución pero visibles al programador, los comentarios ayudan brindando información importante sobre el programa y diferentes funcionalidades.

- b) El objeto usado para imprimir información en pantalla es std::cout.

cout manipula directamente la corriente de salida estándar.

- c) El enunciado de C++ que toma una decisión es un condicional.

Los condicionales en C++ toman decisiones en base al cumplimiento de una condición.

- d) La mayoría de cálculos normalmente se realizan mediante enunciados de asignación.

Se debe a que, cuando realizamos un cálculo, queremos que el resultado no se pierda y podamos usarlo en otra parte del código.

- d) El objeto std::cin ingresa valores desde el teclado.

Opuesto a cout, cin maneja las entradas en el flujo estándar de entrada, en estos casos por teclado.

2.9 Escribe un enunciado o línea que cumpla cada una de las siguientes tareas:

- a) Que imprima el mensaje "Bienvenido al calculador de máximo ritmo cardíaco".

```
std::cout << "Bienvenido al calculador de MRC \n";
```

- b) Asigna el producto de las variables b y c a la variable a.

```
int a = b * c;
```


c) Declara que el programa ayude a calcular el MRC de una persona

// El programa calcula el máximo ritmo cardíaco

d) Ingresa tres variables 'enteros' desde el teclado a, b y c

Int :: cin >> a >> b >> c;

2.10 Indica cual de las siguientes es verdadera o falsa. Explica la respuesta

a) C++ es un lenguaje 'case-sensitive'

Verdadero: C++ trata a variables y caracteres distintos, mayúscula o minúscula

b) Las siguientes variables tienen nombres válidos: findSum2, _findSum2, find_Sum2, _2findSum, 2_findSum, hello, abcOne, var1.x y z.

Falso: Aunque poco convencional, una variable si puede empezar con '-', además de 'A-Z' y 'a-z' sin embargo no puede empezar con números ni tener puntos en medio por lo que son inválidos:

• 2_findSum y • var1.x y z

c) El enunciado "cout >> "hello world";" es un típico ejemplo de salida a una pantalla

Falso: Aunque "hello world" sea el ejemplo más básico usado hoy en día para aprender programación la sintaxis del ejemplo es errónea, siendo el operador ">>" de entrada y no de salida

d) Las siguientes variables tienen nombres inválidos: -1a, 1-a, a-1, \$1a.

Falso: a-1 es la única variable de nombre válido.

2.11 Rellena las espacios en blanco para las enunciados:

a) ¿Que operación aritmética se encuentra al mismo nivel de prioridad que la suma? Substracción

En las operaciones aritméticas las operación de adición y substracción comparten el último nivel en la jerarquía.

b) Cuando una expresión aritmética contiene menos y una multiplicación ¿cual se opera primero? Multiplicación

En la jerarquía $*$, $/$, $\%$ están primero, por lo que si no hay paréntesis se opera la multiplicación primero.

c) Cuando un nuevo valor es puesto en una locación de memoria el proceso es de asignación

Cuando se usa el operador '=' se realiza la asignación de un valor a una variable.

2.12 ¿Que, si algo, se imprime cuando las siguientes enunciados se realizan? Si no imprime nada, la respuesta es "nada".
Asumiendo que $x=2$ y $y=3$

a) `cout << y << " " << x;`

| 3 2

b) `cout << (x+y);`

| 5

c) `cout << "x es igual a " << x;`

| x es igual a 2

d) `cout << "y = " << y;`

| y = 3

e) $z = x * y;$

"nada"

f) $z = z - y$

"nada"

g) $\text{cout} \ll x \ll "\\ " \ll y;$

2 \ 3

h) $// \text{cout} \ll x \ll y;$

"nada"

i) $\text{cout} \ll x \ll "\\n" \ll y;$

2
3

2.13 ¿Cuál de las siguientes enunciaci3n contiene variables cuyos valores han sido reemplazados?

a) $\text{cout} \ll x \ll y;$ X

cout es solo operador de salida NO han sido reemplazados

b) $i = i + j;$ ✓

Reemplaza 'i' al asignarle un nuevo valor, SI ha sido reemplazado

c) $\text{cout} \ll "x";$

Al igual que a), NO ha sido reemplazado

d) $\text{cin} \gg x;$ ✓

El operador de entrada busca asignar un valor a x, por lo que SI, ha sido reemplazado.

2.14 Dada la ecuación algebraica $y = ax^3 + 7$, cual de los siguientes, si hay alguno, son enunciados de C++ correctos para la ecuación

a) $y = a * b * x * x + 7$

Ecuación Deseada

$$y = ax^3 + 7$$

Equivalente Algebraico de a)

$$y = abx^2 + 7$$

No es correcta

b) $y = a * b * x * (x + 7)$

Ecuación Deseada

$$y = ax^3 + 7$$

Equivalente Algebraico de b)

$$y = abx^2 + 7abx$$

No es correcta

c) $y = (a * x) * b * x * (x + 7)$

Ecuación Deseada

$$y = ax^3 + 7$$

Equivalente Algebraico de c)

$$y = abx^3 + 7abx$$

No es correcta

d) $y = (a * x) * b * x + 7$

Ecuación Deseada

$$y = ax^3 + 7$$

Equivalente Algebraico de d)

$$y = ax^2b + 7$$

No es correcta

e) $y = a * (x * x * b) + 7$

Ecuación Deseada

$$y = ax^3 + 7$$

Equivalente Algebraico de e)

$$y = ax^2b + 7$$

No es correcta

f) $y = a * b * (x * x + 7)$

Ecuación Demanda

$$y = ax^3 + 7$$

Equivalente algebraico de F)

$$y = abx^2 + ab \cdot 7$$

No es correcta

Ninguna es correcta ya que todas incluyen 'b', si b fuera x serían correctas a), d) y e)

2.15 Indica el orden de evaluación de los operadores en cada uno de los enunciados de C++ y muestra el valor de x después de ser realizado.

a) $x = 3 + 3 * 4 / 2 - 2;$

① $3 * 4 = 12$

② $12 / 2 = 6$

③ $3 + 6 = 9$

④ $9 - 2 = 7 //$ $x = 7$

b) $x = 4 \% 2 + 2 * 4 - 2 / 2;$

① $4 \% 2 = 0$

② $2 * 4 = 8$

③ $2 / 2 = 1$

④ $0 + 8 = 8$

⑤ $8 - 1 = 7 //$ $x = 7$

c) $x = (2 * 4 * 2 + (9 * 3 / 3));$

① $(9 * 3 / 3)$

① $9 * 3 = 27$

② $27 / 3 = 9$

② $(2 * 4 * 2 + 9)$

① $2 * 4 = 8$

② $8 * 2 = 16$

③ $16 + 9 = 25 //$ $x = 25$

2.10 e) Los paréntesis pueden ser usados para agrupar expresiones

Verdadero: Similar al algebra, se pueden agrupar con ()