



UNIVERSIDAD CATÓLICA
de Colombia



UNIVERSIDAD CATÓLICA
de Colombia
Vigilada Mineducación

Procedural Programming

Basic Concepts

Diego Alberto Rincón Yáñez
darincon@ucatolica.edu.co



Historia...

- Para realizar un algoritmo en un computador se utilizan diversos lenguajes de programación.



Un Algoritmo es...

- Conjunto **ordenado y finito** de operaciones que permite hallar la solución de un problema.
(Real Academia de la Lengua Española)



Especificando el Lenguaje

- Entramos a un lenguaje llamado C++
- Es un lenguaje Orientado a Objetos, pero soporta programación secuencial.
- En este curso aprenderemos programación **secuencial**



¿Qué es la programación Secuencial?



Programación Secuencial

- Programación basada en pasos
 - (Muy similar a los diagramas de flujo)
- No se inicia un paso hasta no finalizar el otro
- En el código, cada línea se ejecuta antes de ejecutar la siguiente.



Ejemplo de pasos Secuenciales



1. Desconfianza



2. Excitación



3. Sorpresa



4. Entusiasmo



5. Amor



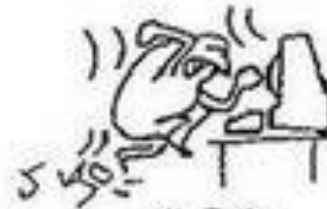
6. Desilusión



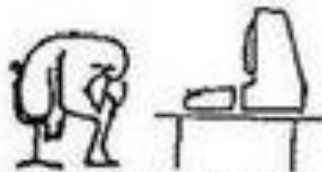
7. Temor



8. Horror



9. Furia



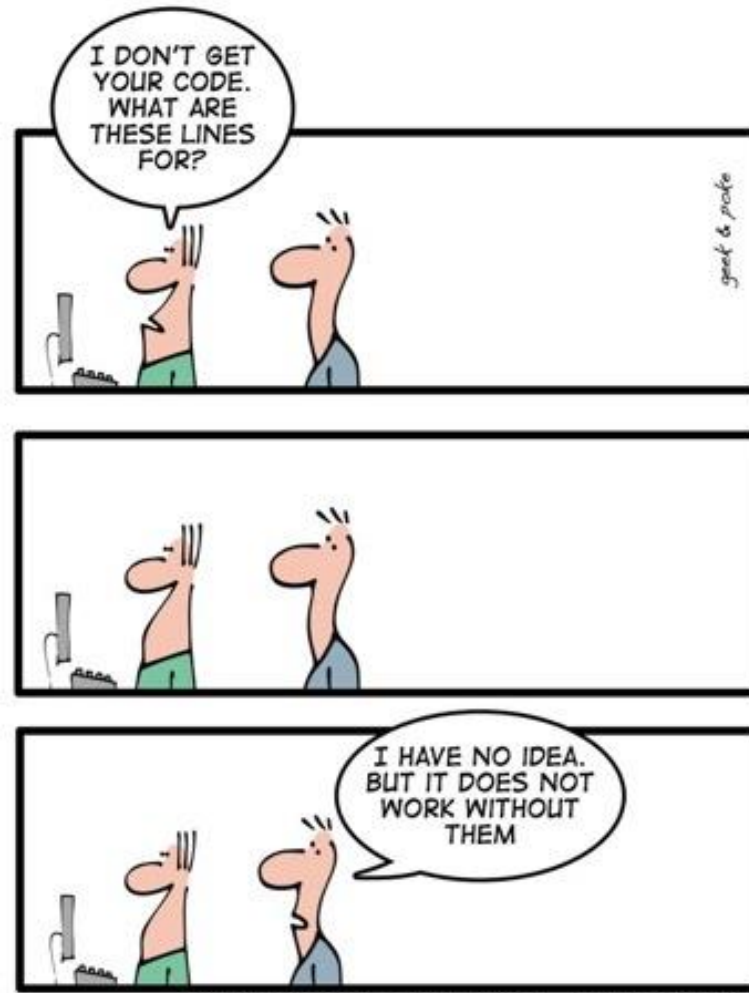
10. Frustración



11. El final



Lo hemos hecho hasta ahora!



THE ART OF PROGRAMMING - PART 2: KISS



Lenguajes de Programación

- Un computador es una máquina muy rápida y muy obediente, hace lo que sea que se le indique (dentro de la razón)
- Sin embargo aprender a hablar “computador” es extremadamente complicado y no natural



Lenguajes de Programación

- Difícilmente un computador entiende un idioma natural como Inglés o Español, y de hacerlo, lo hacen mediante un programa que lo hace, mas no por ellos mismos.
- Los lenguajes de programación son una especie de acuerdo entre las computadoras y el hombre, un lenguaje común que ambas partes puedan “hablar”



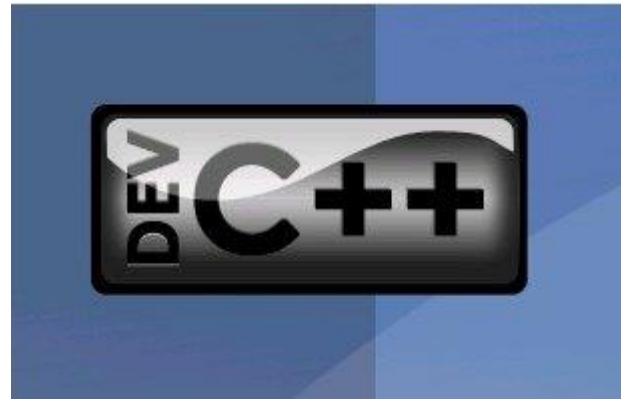
Lenguajes de Programación

- Algunos de esos lenguajes de programación son:
 - C,C++, Java, Ruby, Python, PHP, etc.



Herramienta de Programación

- DEV C++



Para Descargar

- <http://www.bloodshed.net/devcpp.html>



Tipos de Datos

- Entero
- Real (Flotante)
- Caracter
- Cadena (string)
- Booleano

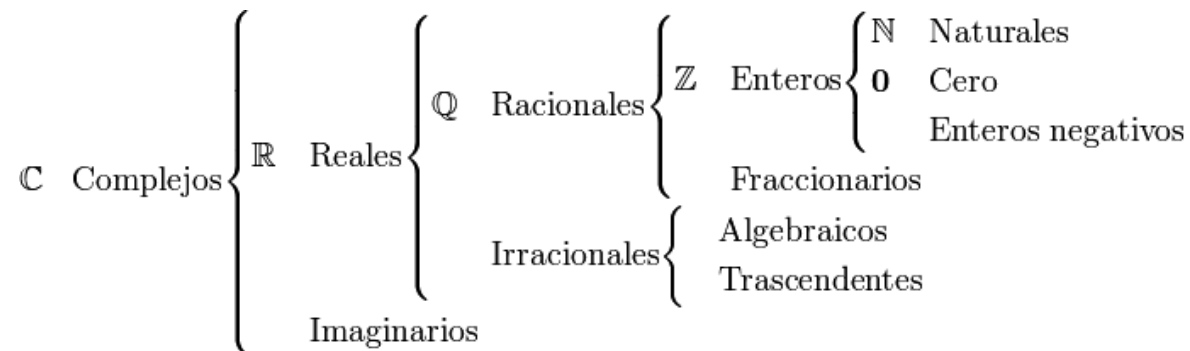
C++

- int
- float
- char
- String
- bool



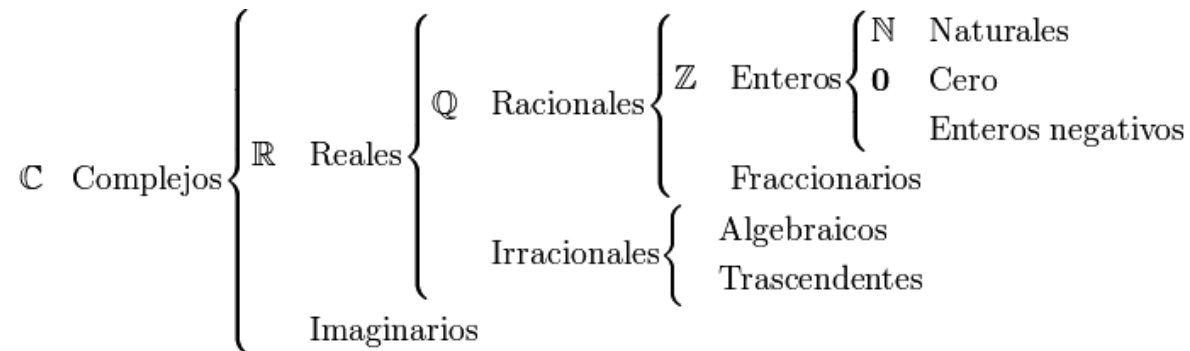
Tipos de Datos: Entero

- Entero:
 - Cantidades indivisibles (Ej. Personas)
 - Resultados naturales entre sumas y restas
 - Parte entera de una división:
 - Ej. $5/2 = 2.5$
 - Un numero entero almacenará el 2

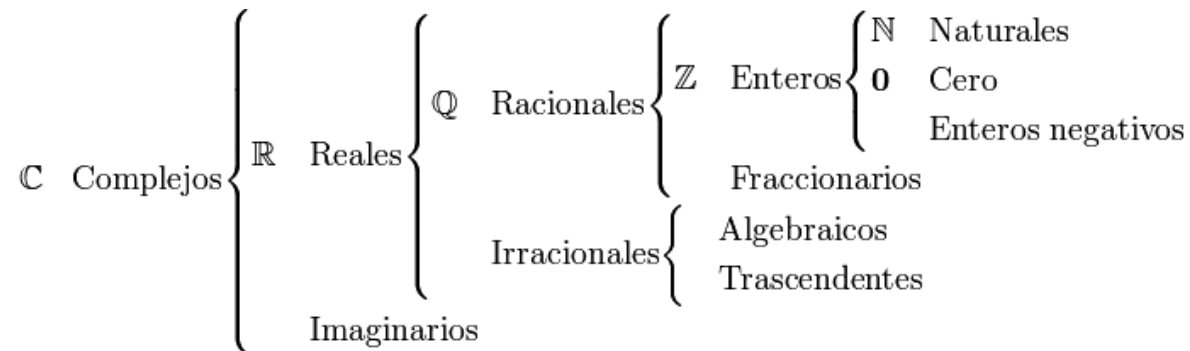
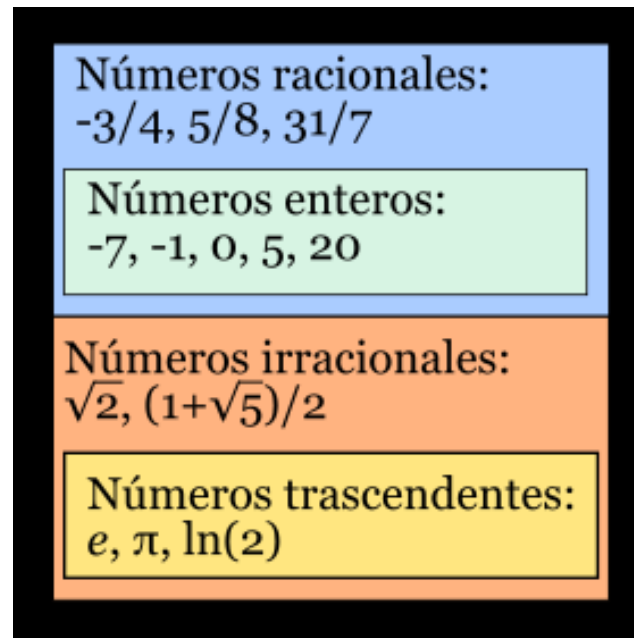


Tipos de Datos: Real (Flotante)

- Real:
 - Cantidades divisibles
 - División completa
 - Ej. $5/2 = 2.5$
 - Un numero flotante almacenará el 2.5



Tipos de Datos: Real (Flotante)



Tipos de Datos: Caracter

- Caracter:
 - Representa una (y sólo una) letra o caracter
 - Ej. 'c', 'C', 'a', 'A', '0', '~'
 - Se representan generalmente con comillas sencillas ('<caracter>')
 - Puede ser un número entero....
 - Ej. 99,67,97,65,48,126
 - Los números enteros almacenados generarán como salida un caracter
 - Ej. Un tipo de dato caracter con valor 64, genera como caracter de salida '@'



Tipos de Datos: Caracter

- Caracter:
 - Son los números que ponemos con la tecla “ALT”!
 - Usualmente son basados en la tabla de valores ASCII
 - Cuidado! Una cosa es el valor que obtiene y otro diferente el valor que representa

• Ej.

Valor entero	Representa
54	'6'
6	'♠'
164	'ñ'



Tipos de Datos: Caracter

- Caracter:
 - Existen caracteres especiales (se diferencian por incluir un carácter adicional ‘\’ que no se almacena)

Modo de Asignación	Representa
<code>'\t'</code>	Tabulación
<code>'\n'</code>	Nueva Línea
<code>'\"'</code>	<code>'\"'</code>



Tipos de Datos: Cadena (string)

- Cadena de Caracteres:
 - Representan una cadena de caracteres (que pueden ser también números)
 - Se representan entre comillas dobles (“<Cadena>”)
 - Pueden existir cadenas de una sola letra
 - Ej. “a”
 - Se diferencian de los caracteres porque es posible agregarle otra letra y no puede asignársele un entero



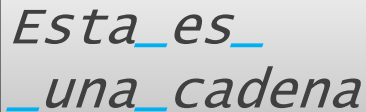
Tipos de Datos: Cadena (string)

- Cadena de Caracteres:
 - Puede contener caracteres especiales
 - Ej. “Esta es \n una cadena” generará



*Esta_es
una_cadena*

- Ej. “Esta es \n una cadena” generará



*Esta_es_
_una_cadena*



Tipos de Datos: Booleano

- Booleano
 - Almacena valores: Verdadero o Falso
 - Usualmente se representa como 0 (Valor Falso) o un numero diferente de 0 (valor verdadero)



Declaración de variables

- En General:
 - Una variable debe tener un nombre único
 - Un nombre de variable no puede contener espacios, iniciar con un número o contener caracteres especiales (#%&@ ...)
 - Un nombre de variable puede mezclar mayúsculas y minúsculas
 - Una variable no puede tener el mismo nombre que una **palabra reservada**
 - Una variable **no debe** iniciar con letra mayúscula, **aunque esto no genere error**



Declaración de variables

- Recomendaciones:
 - Siempre declarar las variables al inicio del programa
 - Se indica el tipo de dato seguido del nombre de la variable y (opcional) un valor inicial
 - Cuando un lenguaje es “case sensitive” es diferente una variable con minúsculas o mayúsculas
 - Ej. `int` hola
`int` hOla = 5



Declaración de variables

- Variables Correctas:
 - `int` mientero
 - `float` miReal
 - `string` mlcAdEnA123
 - Etc..
- Variables incorrectas:
 - `int` entero
 - `float` 2real
 - `string` &%#@
 - `char` mi caracter



Operadores Numéricos

Operador	Representación	Ejemplo
Suma	+	$4+2=6$
Resta	-	$4-2=2$
Multiplicación	*	$4*2=8$
División	/	$4/2=2$
Módulo	%	$4\%2 = 0$
		$4\%3 = 1$

El módulo es el residuo de la división, no puede haber un residuo \geq al divisor

$$4 \overline{) 2 }$$

$$0 \overline{) 2}$$

$$4 \overline{) 3 }$$

$$1 \overline{) 1}$$



Operadores de asignación

Operador	Representación	EJEMPLO
Asignación	=	a = 3
	←	a ← 3

Las asignaciones se realizan de la siguiente forma:

Lo que está en la **derecha del operador** se asigna a lo que está **en la izquierda**



Asignación de Variables

- Para asignar una variable se utilizan el operador de asignación (=)
- En C deben declararse las variables después de la palabra “main(){}”, de la siguiente forma:
<tipoDato> <nombre>
 - Ej:
`int unentero;`
`string unacadena;`
`float unreal;`



Conversión código inicial/final

- C++

```
#include <iostream>  
using namespace std;  
int main ()  
{  
  
    return 0;  
}
```



Conversión Lectura/Escritura

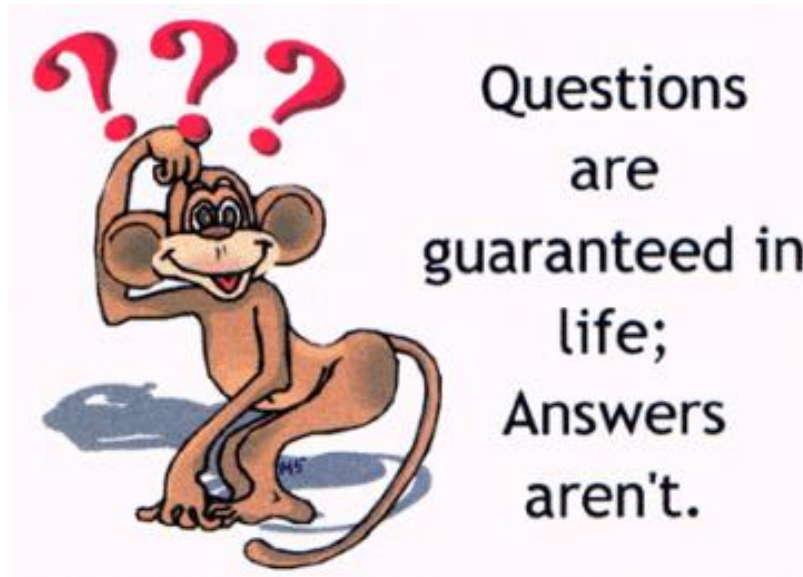
- C++
 - `cout << x << "y" << z;`
 - `cin >> x;`



Leer datos del Usuario

- La función `cin>>valor;` almacena en la variable *valor* (previamente declarada), y es equivalente a hacer el operador = con lo que el usuario ingrese, en este caso, **lo que ingrese el usuario debe obligatoriamente ser del mismo tipo de dato de la variable que lo lee** (el tipo de dato se le dio en la declaración)





¿Questions?

Diego Alberto Rincón Yáñez MCSc.
Twitter: @d1egoprog.

