



UNIVERSIDAD CATÓLICA  
de Colombia



UNIVERSIDAD CATÓLICA  
de Colombia

Vigilada Mineducación

# Procedural Programming

## Basic Concepts

Diego Alberto Rincón Yáñez  
darincon@ucatolica.edu.co



# ¿Qué tipos de dato tendríamos?

- Numero de hermanos:

- -----

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_



# ¿Qué tipos de dato tendríamos?

- Numero de hermanos:

- -----

**INT**

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**



# Podríamos Pasar los formularios por en un grupo...

<ul style="list-style-type: none"><li>• Escriba su Nombre:</li><li>• -----</li></ul> <p><b>STRING</b></p>	<ul style="list-style-type: none"><li>• Escriba su estatura</li><li>• -----</li></ul> <p><b>FLOAT</b></p>
<ul style="list-style-type: none"><li>• Numero de hermanos:</li><li>• -----</li></ul> <p><b>INT</b></p>	<ul style="list-style-type: none"><li>• Es Usted Casado?</li><li>• Si ____</li><li>• No ____</li></ul> <p><b>BOOL</b></p>

# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**



# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

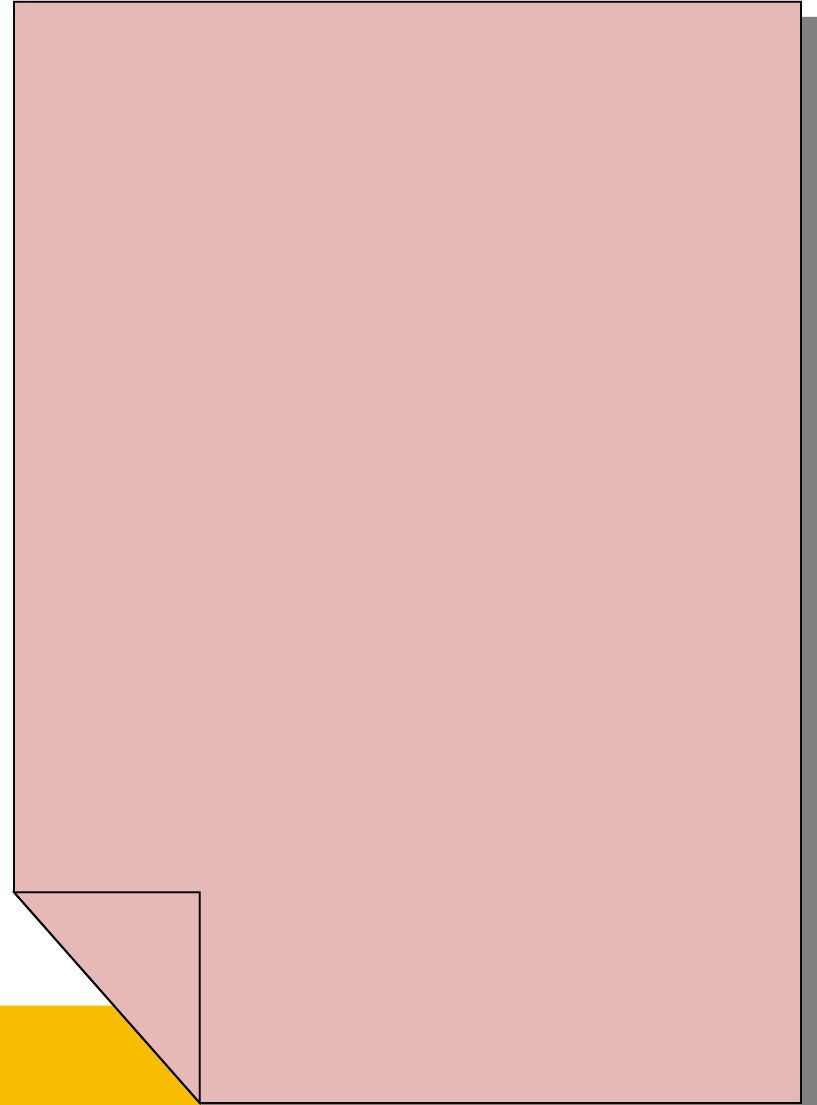
**INT**

- Es Usted Casado?

- Si    \_\_\_

- No    \_\_\_

**BOOL**





# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos

- -----

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_



# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos

- -----

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_

# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos

- -----

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_

# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos

- -----

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_

# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos

- -----

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_

# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos

- -----

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_

# Ejemplo

- Escriba su Nombre:

- -----

**STRING**

- Escriba su estatura

- -----

**FLOAT**

- Numero de hermanos:

- -----

**INT**

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

- Escriba su Nombre:

- ----- **STRING**

- Escriba su estatura

- ----- **FLOAT**

- Numero de hermanos

- ----- **INT**

- Es Usted Casado

- Si \_\_\_\_

- No \_\_\_\_

**BOOL**

# Ejemplo

## Estructura

- Escriba su Nombre:
- ----- **STRING**

- Escriba su estatura
- ----- **FLOAT**

- Numero de hermanos
- ----- **INT**

- Es Usted Casado
- Si \_\_\_\_
- No \_\_\_\_ **BOOL**



# En Código



# Por separado

- Escriba su Nombre:

- -----

- Escriba su estatura

- -----

- Numero de hermanos:

- -----

- Es Usted Casado?

- Si \_\_\_\_

- No \_\_\_\_

- ...

string nombre;

float estatura;

int numHermanos;

bool casado;

...



# Todo Junto

- Escriba su Nombre:
- -----
- Escriba su estatura
- -----
- Numero de hermanos
- -----
- Es Usted Casado
- Si \_\_\_\_
- No \_\_\_\_

```
• struct Formulario{  
  string nombre;  
  float estatura;  
  int numHermanos;  
  bool casado;  
};  
// firmas de  
funciones  
int main(){
```



# Todo Junto

TIPO DE DATO!

- Escriba su Nombre:
- -----
- Escriba su estatura
- -----
- Numero de hermanos
- -----
- Es Usted Casado
- Si \_\_\_\_
- No \_\_\_\_

```
• struct Formulario{  
    string nombre;  
    float estatura;  
    int numHermanos;  
    bool casado;  
};  
  
// firmas de  
funciones  
int main(){
```



# Todo Junto

<ul style="list-style-type: none"><li>• Escriba su Nombre:</li><li>• -----</li></ul>	<b>STRING</b>
<ul style="list-style-type: none"><li>• Escriba su estatura</li><li>• -----</li></ul>	<b>FLOAT</b>
<ul style="list-style-type: none"><li>• Numero de hermanos</li><li>• -----</li></ul>	<b>INT</b>
<ul style="list-style-type: none"><li>• Es Usted Casado</li><li>• Si ____</li><li>• No ____</li></ul>	<b>BOOL</b>
<b>FORMULARIO</b>	

TIPO DE DATO!

```
• struct Formulario{  
    string nombre;  
    float estatura;  
    int numHermanos;  
    bool casado;  
};  
  
// firmas de  
funciones  
int main(){
```



# Para usarlos



```
struct Formulario {  
    string nombre;  
    float estatura;  
    int numHermanos;  
    bool casado;  
};  
//funciones...  
int main(){  
    Formulario unformulario, otroformulario;  
    unformulario.nombre = "Pedro";  
    otroformulario.nombre="Juan";  
  
    unformulario.numHermanos = 1;  
    otroformulario.numHermanos=3;  
  
    // Etc...  
}
```





```
struct Formulario {  
    string nombre;  
    float estatura;  
    int numHermanos;  
    bool casado;  
};  
//funciones...  
int main(){  
    Formulario unformulario, otroformulario;  
    unformulario.nombre = "Pedro";  
    otroformulario.nombre = "Juan";  
  
    unformulario.numHermanos = 1;  
    otroformulario.numHermanos = 3;  
  
    // Etc...  
}
```



```
struct Formulario {  
    string nombre;  
    float estatura;  
    int numHermanos;  
    bool casado;  
};  
  
//funciones...  
  
int main(){  
    Formulario unformulario, otroformulario;  
    unformulario.nombre = "Pedro";  
    otroformulario.nombre = "Juan";  
  
    unformulario.numHermanos = 1;  
    otroformulario.numHermanos = 3;  
  
    // Etc...  
}
```



# Estructuras

- Una estructura es un tipo de dato definido por el usuario
- Es decir que se pueden crear nuevos tipos de dato!
- Los nuevos tipos de dato se basan en los tipos de dato existentes.



# Estructura

```
struct <NombreEstructura>{  
    <tipocampo1> <nombrecampo1>;  
    <tipocampo2> <nombrecampo2>;  
    <tipocampo3> <nombrecampo3>;  
};
```

- Inicia con la palabra reservada **struct** y finaliza con un **;**
- **El nombre de la estructura inicia en mayuscula!**
- Usualmente se declaran antes del main



# ¿Qué son los campos?

- Los campos pueden ser cualquier tipo de dato que hemos visto.
- Un Ejemplo de una estructura puede ser:

```
struct Estudiante{  
    string nombre;  
    int edad;  
    float notas[6];  
}
```



# Pero eso es un tipo de datos

- Como tipo de dato, se necesita declarar una variable para poder usarla, un ejemplo de esto sería:

```
#include <cstdlib>
#include <iostream>

using namespace std;

struct Estudiante{
    string nombre;
    int edad;
    float notas[6];
};

int main(int argc, char *argv[])
{
    int a,b,c;
    Estudiante pepe,juan,pika;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```



# Pero eso es un tipo de dato

```
#include <cstdlib>
#include <iostream>
```

```
using namespace std;
```

```
struct Estudiante{
    string nombre;
    int edad;
    float notas[6];
};
```

Definición de la estructura  
(o nuevo “tipo de dato”)

```
int main(int argc, char *argv[])
{
```

```
    int contador, numero, pos;
    Estudiante pepe, juan, jose;
```

Variables

Tipo de dato

```
    system("PAUSE");
    return EXIT_SUCCESS;
```

```
}
```





# Pero eso es un tipo de dato

```
#include <cstdlib>
#include <iostream>
```

```
using namespace std;
```

```
struct Estudiante{
    string nombre;
    int edad;
    float notas[6];
};
```

Definición de la estructura  
(o nuevo “tipo de dato”)

```
int main(int argc, char *argv[])
{
```

```
    int contador, numero, pos;
    Estudiante pepe, juan, jose;
```

Variables

Tipo de dato

```
    system("PAUSE");
    return EXIT_SUCCESS;
```

```
}
```



# ¿Como se accede a los campos?

```
#include <cstdlib>
#include <iostream>

using namespace std;

struct Estudiante{
    string nombre;
    int edad;
    float notas[5];
};
```

```
int main(int argc, char *argv[])
{
    int        contador,numero,pos;
    Estudiante pepe,juan,jose;

    pepe.nombre = "Pepe";
    pepe.edad = 18;
    pepe.notas[0]=3.0f;
    pepe.notas[1]=3.5f;
    pepe.notas[2]=2.8f;
    pepe.notas[3]=3.7f;
    pepe.notas[4]=4.0f;

    cout<<"Nombre: "<<pepe.nombre<<endl;
    cout<<"Edad: "<<pepe.edad<<endl;
    cout<<"Notas: ";
    for(int i=0;i<5;i++){
        cout<<"\t"<<i+1<<": "<<pepe.notas[i]<<endl;
    }

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# ¿Como se accede a los campos?

```
#include <cstdlib>
#include <iostream>

using namespace std;

struct Estudiante{
    string nombre;
    int edad;
    float notas[5];
};
```

```
int main(int argc, char *argv[])
{
    int contador, numero, pos;
    Estudiante pepe, juan, jose;

    pepe.nombre = "Pepe";
    pepe.edad = 18;
    pepe.notas[0] = 3.0f;
    pepe.notas[1] = 3.5f;
    pepe.notas[2] = 2.8f;
    pepe.notas[3] = 3.7f;
    pepe.notas[4] = 4.0f;

    cout<<"Nombre: "<<pepe.nombre<<endl;
    cout<<"Edad: "<<pepe.edad<<endl;
    cout<<"Notas: ";
    for(int i=0; i<5; i++){
        cout<<"\t"<<i+1<<": "<<pepe.notas[i]<<endl;
    }

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# Estructuras y funciones

- Se maneja igual a cualquier tipo de dato...
- Las estructuras deben definirse ANTES de las firmas de las funciones

```
#include <cstdlib>
#include <iostream>
using namespace std;
//Definición de Estructuras
struct Estudiante{
    string nombre;
    int edad;
    float notas[5];
};
//Definición de Funciones
Estudiante llenarEstudiante (Estudiante est);
void imprimirEstudiante(Estudiante est);
```



# Estructuras y funciones

- Se maneja igual a cualquier tipo de dato...
- Las estructuras deben definirse ANTES de las firmas de las funciones

```
#include <cstdlib>
#include <iostream>
using namespace std;
//Definición de Estructuras
struct Estudiante{
    string nombre;
    int edad;
    float notas[5];
};
//Definición de Funciones
Estudiante llenarEstudiante (Estudiante est);
void imprimirEstudiante (Estudiante est);
```



# Estructuras y funciones

- En la definición de la función....

```
Estudiante llenarEstudiante (Estudiante est) {
    est.nombre = "Pepe";
    est.edad = 18;
    est.notas[0]=3.0f;
    est.notas[1]=3.5f;
    est.notas[2]=2.8f;
    est.notas[3]=3.7f;
    est.notas[4]=4.0f;
    return est;
}

void imprimirEstudiante(Estudiante est) {
    cout<<"Nombre: "<<est.nombre<<endl;
    cout<<"Edad: "<<est.edad<<endl;
    cout<<"Notas: ";
    for(int i=0;i<5;i++){
        cout<<"\t"<<i+1<<": "<<est.notas[i]<<endl;
    }
}
```



# Estructuras y funciones

- En la función principal...

```
int main(int argc, char *argv[])
{
    Estudiante pepe,juan,jose;
    pepe = llenarEstudiante (pepe);
    imprimirEstudiante(pepe);
    system("PAUSE");
    return EXIT_SUCCESS;
}
```





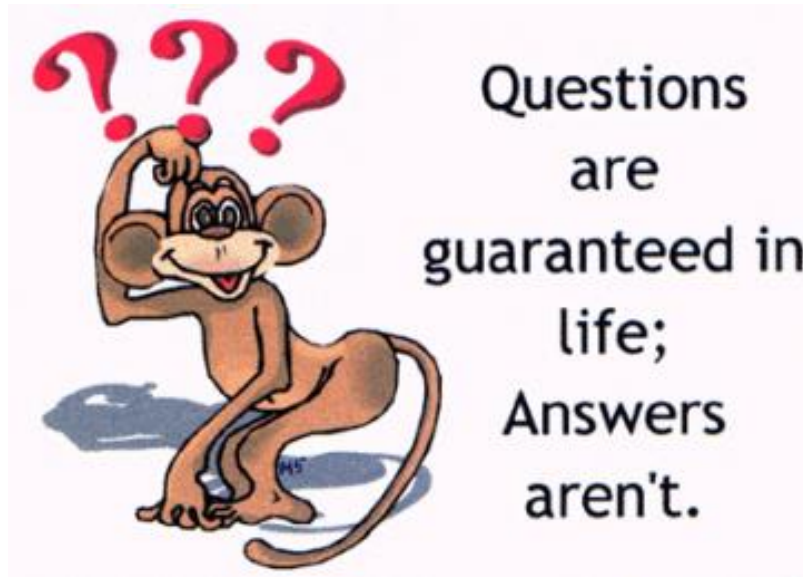
# Arreglos de Estructuras

- Se pueden declarar arreglos de cualquier dimensión de estructuras:

```
struct Prueba{
    int dato1;
    string dato2;
    bool dato3;
};

int main(int argc, char *argv[])
{
    Prueba miPrueba[10];
    for(int i=0;i<10;i++){
        miPrueba[i].dato1=i;
        miPrueba[i].dato2="Yo";
        miPrueba[i].dato3=(i%2==0);
    }
    for(int i=0;i<10;i++){
        cout<<miPrueba[i].dato1<<endl;
        cout<<miPrueba[i].dato2<<endl;
        cout<<miPrueba[i].dato3<<endl<<endl;
    }
}
```





# ¿Questions?

Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

