

 **Informatique S1**  
**Programmation C**

- *Objectifs de la séance*
  - structure générale d'un programme en C
- *Concepts*
  - Structure générale
  - Mots-clés
  - Variables
  - Opérateurs arithmétiques
  - Commentaires
- Exemples avec *printf* et *scanf*

---

---

---

---

---

---

---

---

 **Structure général**

```

2  #include <stdio.h>
3
4  /* mon premier programme en c */
5  int main (int argc, char argv[]) {
6      /* age de l'utilisateur */
7      int age;
8      int annee;
9
10     /* initialisation des variables */
11     age=0;
12     annee=0;
13
14     printf ("Indiquer votre annee de naissance: ");
15     scanf ("%d",&annee);
16
17     age = 2008 - annee;
18
19     printf ("Vous avez %d ans\n", age);
20
21 }
22
  
```

Annotations: Directives (import des bibliothèques) points to line 2; Commentaires points to line 4; Déclarations points to lines 7-8; Instructions d'attribution points to lines 11-12; Fonction points to line 21.

---

---

---

---

---

---

---

---

 **Mots-clés**

- Mots réservés
- Mots reconnues par le compilateur
- Chacune a une signification particulière

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

---

---

---

---

---

---

---

---



## Variables



- **Définition :**
  - Une entité qui contient une information
  - Les variables sont stockées dans la mémoire
- **Caractéristiques :**
  - nom → identifiant
  - valeur
  - type
    - Le type indique l'ensemble des valeurs que peut prendre la variable

---

---

---

---

---

---

---

---



## Les types des données

- Les principaux types acceptés en langage C :
  - Entiers
    - int (± 16 bits)
    - short (± 8 bits)
    - long (± 32bits)
  - Réels
    - float
    - double
  - Caractères
    - char
  - Signed / unsigned
  - void

char	float	int
long	short	signed
unsigned	void	

---

---

---

---

---

---

---

---



## Actions sur les variables

- Que peut-on faire avec une variable ?
  - Déclaration → définir la variable
  - Lire (obtenir) sa valeur → regarder son contenu
  - Affecter une (nouvelle) valeur → mettre une (nouvelle) information
    - Opérateur d'attribution : « = »

---

---

---

---

---

---

---

---

**Déclaration d'une variable**

- **Format :**  
 – type identifiant [ , identifiant... ] [ = valeur initial ] ;
- **Exemples :**  

```
int i, j, k;
char lettre;
float reel01 = 1.25;
```

Diagramme de la syntaxe :

```

    optionnel
    |
    +-----+
    | type identifiant [ , identifiant... ] [ = valeur initial ] ; |
    +-----+
    
```

Annotations :

- On sépare les identifiants par les « , »
- type
- identifiant
- Identifiant  
 lettre [ lettres, chiffres ou \_ ]  
 a10 ~~10a~~  
 var\_int ~~var!~~

---

---

---

---

---

---

---

---

**Attribution**

- **Opérateur d'attribution :** « = »
- **Exemple :**  

```
c = a + b;
```
- **Évaluation**
  - On prend la **valeur** contenue dans la **variable a**
  - On prend la **valeur** contenue dans la **variable b**
  - On additionne (opérateur « + ») ces deux valeurs
  - On met ce résultat dans la **variable c**
- Si **c** avait auparavant une **valeur**, cette dernière est **perdue !**

---

---

---

---

---

---

---

---

**Opérateurs**

- **Opérateurs arithmétiques :**  

```
* /
%
+ -
```

Ordre de priorité ↓

Le résultat d'un opérateur dépend des types des variables !!
- **Exemples :**  

```
int a, b;
a = 2 + 3 * 5;
b = 2 * 5 + 5 % 2;
b = 5 / 2;
```

25 ou 17 ?

---

---

---

---

---

---

---

---

**Exemple**

```

2  #include <stdio.h>
3
4  /* mon premier programme en c */
5
6  int main (int argc, char argv[]) {
7      // age de l'utilisateur
8      int age;
9      int annee;
10
11     /* initialisation des variables */
12     age=0;
13     annee=0;
14
15     printf ("Indiquer votre annee de naissance: ");
16     scanf ("%d",&annee);
17
18     age = 2008 - annee;
19
20     printf ("Vous avez %d ans\n", age);
21
22 }
    
```

---

---

---

---

---

---

---

---

**Entrée & Sortie**

- Entrée formatée : scanf
- Sortie formatée : printf
- Formats
  - %d int
  - %i
  - %f float
  - %e
  - %c char
- Caractères spéciaux
  - \n nouvelle ligne
  - \t tab
  - \\ la « \ »
  - \" le « \" »
  - %% le « % »

---

---

---

---

---

---

---

---