

122

LS1 - Algorithmique/programmation renforcés ;
langage C - Ph. Hunel
2025-2026

Chapitre 8 : Types évolués

122

123

8.1 – Les structures

- ▶ Les tableaux sont des paquets de données de même type.
- ▶ Les structures sont des ensembles de données non homogènes.
- ▶ Les données peuvent avoir des types différents.
- ▶ Les structures sont déclarées ou définies selon le modèle :

LS1 - Algorithmique/programmation renforcés ;
langage C - Ph. Hunel

2025-2026

123

124

8.1 – Les structures

- ▶ Les structures sont déclarées ou définies selon le modèle :

```
struct nom_de_structure_facultatif {  
    la liste des données contenues dans la  
    structure  
} la liste des variables construites selon ce modèle.
```

LS1 - Algorithmique/programmation renforcés ;
langage C - Ph. Hunel

2025-2026

124

125

8.1 – Les structures

- ▶ Exemple :

```
struct etd {  
    char nom[20];  
    char prenom[20];  
    float note_info;  
    float note_phi;  
    float moyenne;  
} Honorer;
```

LS1 - Algorithmique/programmation renforcés ;
langage C - Ph. Hunel

2025-2026

125

8.1 – Les structures

- ▶ `etd` est un nom de modèle de structure
- ▶ `Honorier` est un objet de type `struct etd.`
- ▶ Les différentes parties de la structure `Honorier` sont accessibles par :
 - ▶ `Honorier.nom`
 - ▶ `Honorier.prenom`
 - ▶ `Honorier.note_info`
 - ▶ `Honorier.note_phi`
 - ▶ `Honorier.moyenne`

LS1 - Algorithmique/programmation renforcés ;
langage C - Ph. Hunel

2025-2026

126

8.2 – Les unions

- ▶ Les **unions** permettent l'utilisation d'un même espace mémoire par des données de types différents à des moments différents.
 - ▶ Syntaxe de la définition d'une union
- ```
union nom_union {
 type1 nom_champ1;
 type2 nom_champ2;
 ...
 typeN nom_champN;
} variables;
```

LS1 - Algorithmique/programmation renforcés ;  
langage C - Ph. Hunel

2025-2026

127

## 8.2 – Les unions

- ▶ Exemple :

```
union zone {
 int entier;
 float flottant;
} z1;
```

- ▶ Utilisation
  - ▶ z1.entier = 1;
  - ▶ z1.flottant = 2.5;

## 8.3 – Opération sur les structures et les unions

- ▶ L'affectation d'un objet de type structure est réalisé par l'opérateur égal (=)
  - ▶ réalise une affectation membre à membre des membres de l'objet situé à gauche du signe = par ceux de l'expression situé à droite.
  - ▶ **ATTENTION :**
    - ▶ Lorsque l'un des membres est un pointeur, le pointeur est affecté par un autre pointeur.
    - ▶ Les deux membres pointeurs des deux objets différents pointent sur la même donnée.
    - ▶ pas de duplication de la zone pointée.

130

### 8.3 – Opération sur les structures et les unions

- ▶ Les objets de type structure peuvent être passés en argument d'une fonction.
- ▶ Une fonction peut recevoir en argument un pointeur sur un objet de type structure.
- ▶ Une fonction peut retourner un objet de type structure.

L3I - Algorithmique/programmation renforcés ; langage C - Ph.  
Hunel

2025-2026

130

131

### 8.4 – Opérateur `typedef`

- ▶ Le mot-clé **`typedef`** permet de renommer un type.

**`typedef type nouveau_type ;`**

- ▶ permet de simplifier la lisibilité de certaines déclaration ou de rendre un programme plus portable en renommant certains types susceptibles de varier d'un compilateur à l'autre.

- ▶ Exemple :

`typedef unsigned short int usint ;`

- ▶ renomme le type `unsigned short int` en `usint`.

L3I - Algorithmique/programmation renforcés ; langage C - Ph.  
Hunel

2025-2026

131

## 8.5 – exemple de programme

```
#include <stdio.h>

typedef struct ed {
 char nom[20];
 char prenom[20];
 int analyse, info2;
 int *math1;
 float moyenne;
} etudiant;

void affiche(etudiant e) {
 printf("Nom : %s\n", e.nom);
 printf("Prénom : %s\n", e.prenom);
 printf("Note d'analyse : %d\n", e.analyse);
 printf("Note d'informatique 2 : %d\n", e.info2);
 printf("Note de maths 1 : %d\n", *e.math1);
 printf("Moyenne : %.2f\n", e.moyenne);
}

LSI - Algorithmique/programmation renforcés ; langage C - Ph.
Hunel
2025-2026
```

132

## 8.5 – exemple de programme

```
void main() {
 etudiant e1,e2;
 printf("e1 nom : "); scanf("%s",e1.nom);
 printf("e1 prénom : "); scanf("%s",e1.prenom);
 printf("e1 analyse : "); scanf("%d",&e1.analyse);
 printf("e1 info2 : "); scanf("%d",&e1.info2);
 printf("e1 math1 : "); scanf("%d",&e1.math1);
 e1.moyenne = (e1.analyse + e1.info2 + *e1.math1)/3;
 printf("\n---- e1 ----\n");
 affiche(e1);
 e2=e1;
 printf("\n---- e2 ----\n");
 affiche(e2);
 *e2.math1=100;
 strcpy(e2.nom, "LE PROF");
 printf("\n---- e2 après modification de e2 ----\n");
 affiche(e2);
 printf("\n---- e1 ----\n");
 affiche(e1);
}
```

LSI - Algorithmique/programmation renforcés ; langage C - Ph.  
Hunel  
2025-2026

133

## 8.6 – Exercices

- ▶ Définissez un type Point représentant les coordonnées (abscisse et ordonnée) réelles d'un point dans un plan
- ▶ Donnez une fonction qui renvoie dans Point les coordonnées lues au clavier
- ▶ Donnez la fonction qui calcule la distance euclidienne entre deux points