

# Introduction au Langage C,C++

## # Les structures de données

### ■ intérêt

- Rassembler des données hétérogènes caractérisant une entité pour en faire un type utilisateur.

### ■ exemple:

- # point dans l'espace → 3 entiers
- # nœud d'un arbre binaire → 2 adresses vers les fils, 3 entiers pour les données du nœud
- # noms et références des pièces mécaniques constituant une voiture  
→ 600 tableaux de caractères + 600 entiers pour la référence

# Introduction au Langage C,C++

## # Les structures de données

### ■ déclaration

```
struct nom_structure  
{  
    type1 nomchamps1;  
    type2 nomchamps2;  
    .  
    .  
    typeN nomchampsN;  
};
```

# Introduction au Langage C,C++

## \* Les structures de données

### ■ exemple et utilisation

```
struct pt
{
    int x;
    int y;
    int z;
    char nom;
};

main()
{
    struct pt p;
}
```



# Introduction au Langage C,C++

## \* Les structures de données

### ■ simplification de l'écriture

```
struct pt  
{
```

```
    int x;  
    int y;  
    int z;  
    char nom;
```

```
};
```

```
typedef struct pt point;
```

```
main()  
{
```

```
    point p;  
}
```

Notations  
équivalentes

```
typedef struct pt  
{
```

```
    int x;  
    int y;  
    int z;  
    char nom;
```

```
} point;
```

# Introduction au Langage C,C++

## \* Les structures de données

### ■ accès aux données

#### ■ Deux cas de figure

\* On dispose du nom de la variable

■ Accès par: NomVariable.NomChamps

#### \* exemple

```
main()
```

```
{
```

```
    point p1;
```

```
    p1.x = 8; p1.y = 9; p1.z = 10;
```

```
    cout <<p1.x<<p1.y<<p1.z<<'\\n';
```

```
}
```

# Introduction au Langage C,C++

## # Les structures de données

### ■ accès aux données

#### ■ Deux cas de figure

# On dispose de l'adresse de la variable (pointeur)

■ Accès par: NomVariable->NomChamps

#### # exemple

```
main()
```

```
{
```

```
    point *pp1, p1;
```

```
    pp1 = &p1;
```

```
    pp1->x = 8; pp1->y = 9; pp1->z = 10;
```

```
    cout <<pp1->x<<pp1->y<<pp1->z<<'\\n';
```

```
}
```