

集训课-09：一维数组

课程目标

01

数组声明

02

数组初始化

03

获取数组元素

04

数组输入输出

05

总结回顾



Part 1

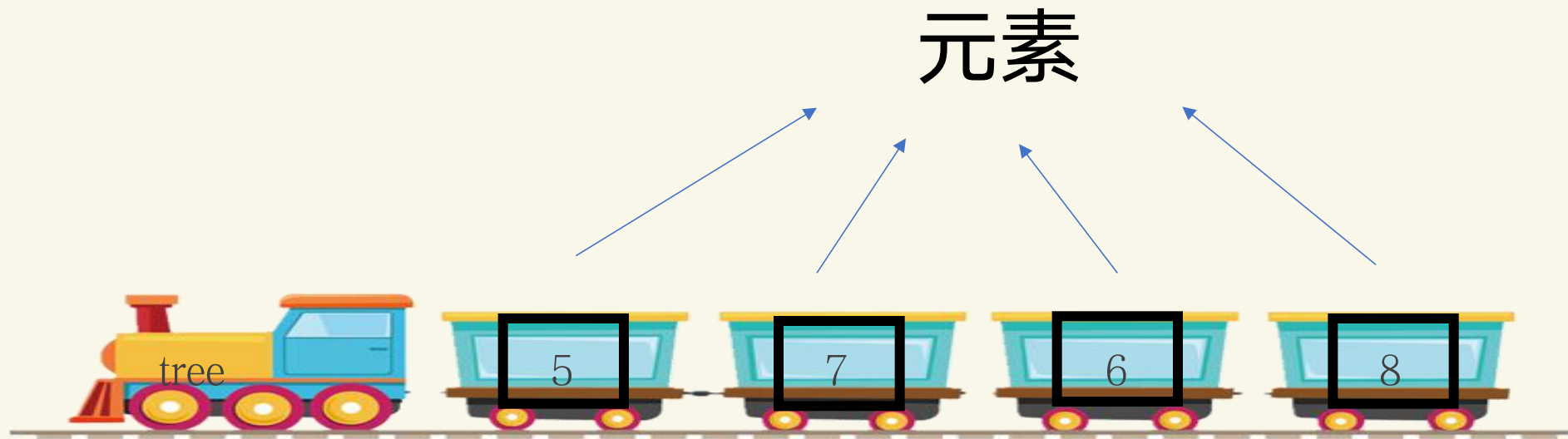
数组声明

数组声明

数组：

一组具有相同数据类型变量的集合

数组声明



数组声明

★ 数据类型 数组名[元素个数];

例如

```
int array[5];
```

```
double arr[6];
```

数组声明

★ 数据类型 数组名[元素个数];

元素个数代表数组中最多存几个元素

数组声明

★ 标识符命名规则：

- ✓ 字母、数字、下划线的组合
- ✓ 不能以数字作为开头
- ✓ 不能是C++关键字
- ✓ 区分大小写
- ✓ 不能冲突和重复
- ✓ 注意：见名知意

数组声明

下列可以作为数组名的是

A、 3apple

B、 num

C、 q@q

数组声明

存储10个人的年龄

```
int age[10];
```

存储5个人的体温

```
double t[5];
```

数组声明

```
int array[10];
```

array数组最多可以存储多少个元素？

A. 9

B. 10

C. 11



Part 2

数组初始化

数组初始化

依次初始化所有元素

```
int tree[5] = {6,7,8,9,10};
```

数组初始化

所有元素全部初始化为0

数据类型 数组名[元素个数] = {};

例如:

```
int tree[5] = {};
```

数组初始化

初始化第一个元素，其他的元素为0

数据类型 数组名[元素个数] = {};

例如:

```
int tree[5] = {1};
```

数组初始化

`int tree[5] = {};`tree数组里面的5个元素分别是？

A. 0 1 2 3 4

B. 1 2 3 4 5

C. 0 0 0 0 0



Part 3

获取数组元素

获取数组元素



数组下标：元素在数组中的位置

获取数组元素



通过数组下标获取数组元素的值

格式：数组名[下标]

```
int tree[5] = {6,7,8,9,10};
```

下标 0 1 2 3 4

```
cout << tree[0]; //获取到的值是6
```

获取数组元素

数组下标从0开始

最后一个下标为数组元素个数-1

获取数组元素

下标必须为自然数

不可以为负数和浮点数

获取数组元素

使用的数组下标**超出**数组的实际下标

会发生数组下标越界，是错误的

获取数组元素

下面数组中6的下标是？

A. 2

B. 1

C. 0

```
int pig[5] = {6,7,8,9,10};
```

获取数组元素

下面程序输出的结果是？

A. 7

B. 8

C. 9

```
int bird[5] = {6,7,8,9,10};
```

```
cout << bird[1];
```



Part 4

数组输入输出

数组输入输出



通过数组下标输出数组元素的值

格式：cout << 数组名[下标];

```
int tree[5] = {};
```

```
//输出下标为0的元素的值
```

```
cout << tree[0];
```

数组输入输出

★ 通过使用数组下标，依次获取数组里面的所有元素

```
int tree[5] = {};  
for (int i = 0; i < 5; i++)  
{  
    cout << tree[i] << " ";  
}
```

下标	树的高度
0	tree[0]
1	tree[1]
...	...
i	tree[i]

数组输入输出

★ 通过使用数组下标，依次输入数组里面的所有元素

```
int tree[5] = {};  
for (int i = 0; i < 5; i++)  
{  
    cin >> tree[i];  
}
```

下标	树的高度
0	tree[0]
1	tree[1]
...	...
i	tree[i]



Part 5

总结回顾

总结回顾

一、基本概念

数组：一组相同数据类型的元素

下标：下标必须为自然数，不可以为负数和浮点数

越界：超出数组存储的范围

总结回顾

二、一维数组输入

```
int array[5];
for (int i = 0 ; i < 5 ; i++)
{
    //输入元素的值，存储到下标为i的数组array中，即array[i]
    cin >> array[i];
}
```

三、一维数组输出

```
int array[5];
for (int i = 0 ; i < 5 ; i++)
{
    //输出下标为i的元素的值，即array[i]
    cout << array[i];
}
```

A horizontal yellow banner with blue triangular ends, centered on a solid blue background. The text "Class is over" is written in blue within the banner.

Class is over