

- ❖ **Array** العادية تحجز مساحة محددة في الذاكرة – سواء استخدمتها كلها أو لا –
- ❖ حجز مساحة في الذاكرة مع عدم استخدامها يؤدي الى بطيء في البرنامج
- ❖ **Array** مع **Pointer** تحجز في الذاكرة ما تحتاجه فقط
- ❖ كل **Type new** لا بد أن تستخدم مقابلها **delete** NamePointer

```

int Num;
cout << "Enter total number of students : ";
cin >> Num;

float* ptr;

// memory allocation of Num number of floats
ptr = new float[Num];

cout << "Enter grades of students." << endl;

for (int x = 0; x < Num; x++)
{
    // Array = [ x + 1 ] رقم العنصر في
    cout << "Student [ " << x + 1 << " ] : ";

    // index = (ptr + x)
    cin >> *(ptr + x);
}

cout << "\n Displaying grades of students." << endl;

for (int i = 0; i < Num; i++)
{
    cout << "Student [ " << i + 1 << " ] : " << *(ptr + i) <<
endl;
}

// ptr memory is released
delete [] ptr;

```