

- ❖ أول عنصر يدخل في **Stack** هو آخر عنصر يخرج من **Stack** ( والعكس بالعكس )
- ❖ إضافة عنصر الى **Stack** يسمى **Push**
- ❖ إزالة عنصر من **Stack** يسمى **Pop** ( يزيل آخر عنصر دخل الى stack أولاً )
- ❖ **Vector** عبارة عن **Stack** هو ( يستخدم لتخزين البيانات والتعامل معها بطريقة معينة )

```
vector<int> vNumbers ;

vNumbers.push_back(10);
vNumbers.push_back(20);
vNumbers.push_back(30);
vNumbers.push_back(40);
vNumbers.push_back(50);

// طباعة حجم ( عدد ) العناصر الموجودة في Stack - بعد تعبئة عناصرها -
cout << "Stack Size : " << vNumbers.size() << endl;

//... إزالة آخر عنصر دخل Stack (50)
vNumbers.pop_back();
vNumbers.pop_back();
vNumbers.pop_back();
vNumbers.pop_back();
vNumbers.pop_back();

// طباعة حجم ( عدد ) العناصر الموجودة في Stack بعد إزالة عناصرها
cout << "Stack Size : " << vNumbers.size() << endl;

// إذا استخدمت pop_back() و Stack لا يوجد به عناصر : البرنامج لا يشتغل error
// للتأكد من أن Stack فاضي استخدم

// إذا Stack في vNumbers ليس فارغ نفذ الشرط 1-
if ( ! vNumbers.empty() )
    vNumbers.pop_back();

// إذا كان عدد العناصر في vNumbers داخل Stack أكبر من 0 نفذ الشرط 2-
if ( vNumbers.size() > 0 )
    vNumbers.pop_back();

// لإزالة جميع العناصر من Stack دفعة واحدة
vNumbers.clear();

// إزالة جميع العناصر باستخدام Ranged Loop
for (int& Number : vNumbers)
{
    vNumbers.pop_back();
}

cout << "Numbers Vector : \n\n";

for (int& Number : vNumbers)
{
    cout << Number << endl;
}
cout << endl;
```