Date:.....

ITEM 16 : Use Same form in new & delete.

→ string * Array = new string [100];

→ delete Array;

Everything looks fine but there is a problem.

How delete works →

① One or more destructors are called for the memory.

② Memory is deallocated.

In the above delete Array gives undefined behaviour since Array is not a single object it an array of object

Delete should know how many time the destructor needs to be called.

① string * Array1 = new string;

delete Array1; // Single obj destructor called.

② string * Array2 = new string [10];

delete[] Array2; // Destructor for 10 objects called. deletes an array of object.

Date:.....

if we use delete [] Array1 , result is undefined.

→ Rule is
If you use [] in a new expression, you must use [] in corresponding delete expression. If you don't use [] in a new expression, don't use [] in matching delete expression.

→ Avoid using typedef for array types.

```
typedef string Add[4];

string *pal = Add;

delete pal ; // undefined
delete[] pal ; // fine
```

Instead we can use vector<string>