

Defines dispatching method

```
class TriangulateCell : public vtkm::worklet::WorkletMapPointToCell  
{
```

Defines how input arrays and structures are interpreted

```
public:  
    typedef void ControlSignature(CellSetIn topology,  
                                  ExecObject tables,  
                                  FieldOutCell<> connectivityOut);
```

```
    typedef void ExecutionSignature(CellShape, PointIndices, _2, _3, VisitIndex);
```

```
    using InputDomain = _1;
```

Specifies domain argument (optional)

```
    using ScatterType = vtkm::worklet::ScatterCounting;
```

Defines how data are  
assigned to threads

```
    template<typename CellShapeTag,  
             typename ConnectivityInVec,  
             typename ConnectivityOutVec>  
    VTKM_EXEC
```

Defines mapping from  
input domain to output  
domain (optional)

```
    void operator()(  
        CellShapeTag shape,  
        const ConnectivityInVec &connectivityIn,  
        const internal::TriangulateTablesExecutionObject<DeviceAdapter> &tables,  
        ConnectivityOutVec &connectivityOut,  
        vtkm::IdComponent visitIndex) const
```

Algorithms are just functions that  
run on a single instance of the input

```
{
```