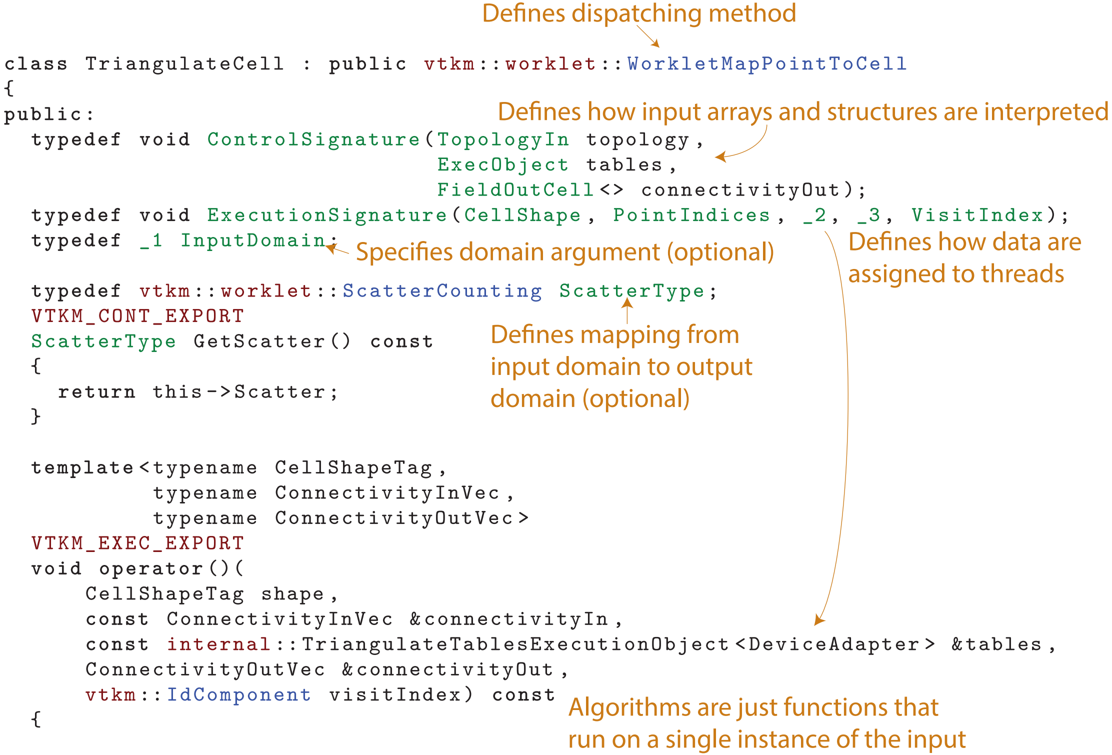

Defines dispatching method

```
class TriangulateCell : public vtkm::worklet::WorkletMapPointToCell
{
public:
  typedef void ControlSignature(TopologyIn topology,
                                ExecObject tables,
                                FieldOutCell<> connectivityOut);
  typedef void ExecutionSignature(CellShape, PointIndices, _2, _3, VisitIndex);
  typedef _1 InputDomain;

  typedef vtkm::worklet::ScatterCounting ScatterType;
  VTKM_CONT_EXPORT
  ScatterType GetScatter() const
  {
    return this->Scatter;
  }

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

Defines how input arrays and structures are interpreted

Specifies domain argument (optional)

Defines how data are assigned to threads

Defines mapping from input domain to output domain (optional)

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