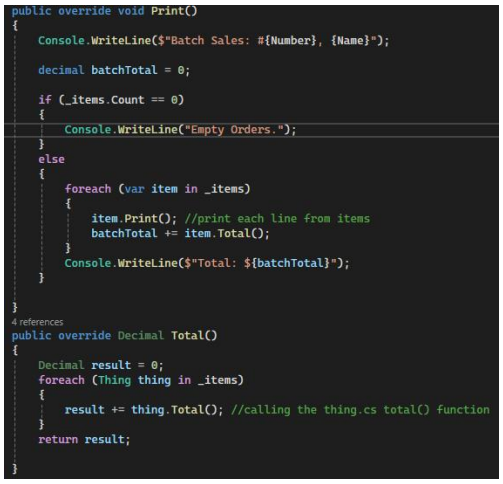
Task 2

1.Describe the principles of polymorphism and how and where it is being used in Task 1?

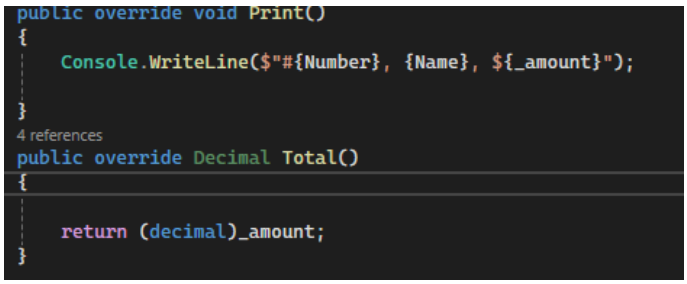
-Polymorphism is a concept that can take on different forms in different classes. The ability to have a different set of behaviour in different classes but same name inherited from mother class. The use of Subclasses and override from Thing to Batch and Transaction

Batch



```
public override void Print()
{
    Console.WriteLine($"Batch Sales: #{Number}, {Name}");

    decimal batchTotal = 0;

    if (_items.Count == 0)
    {
        Console.WriteLine("Empty Orders.");
    }
    else
    {
        foreach (var item in _items)
        {
            item.Print(); //print each line from items
            batchTotal += item.Total();
        }
        Console.WriteLine($"Total: ${batchTotal}");
    }

}
4 references
public override Decimal Total()
{
    Decimal result = 0;
    foreach (Thing thing in _items)
    {
        result += thing.Total(); //calling the thing.cs total() function
    }
    return result;

}
```

Transaction



```
public override void Print()
{
    Console.WriteLine($"#{Number}, {Name}, ${_amount}");

}
4 references
public override Decimal Total()
{

    return (decimal)_amount;
}
```

The use of Polymorphism is what we call each Thing in its list _items to Print(). It's allow many different implementation as a sub class to its mother class. When with Total() it returns the decimal _amount which is total of all the Thing total(). Implement in sales like PrintOrders() methods because each of the Thing weather it is batch or transaction is called to be Print() and Total() are summed to return it's value.

2.What is wrong with the class name Thing? Suggest a better name and reasoning behind your answer.

-Thing doesn't hold any value as it's too vague when naming classes. I would suggest

changing class Thing name to Checkout as the concept could be in a bookstore setting. The reason behind it is because a Checkout can calculate batch, sales and process transaction.

3.What is abstraction and how and where it is being used in Task 1.

Below is all the abstraction used in Task 1, private string _number or _name are private attributes which derived from the Thing class. The properties are similar to it's readonly properties and by performing data hiding with the used of private we are able to hide the stuff we don't want to show which is the application of abstraction itself.

```
private float _amount;
```

```
private string _number;
private string _name;
```

```
private List<Thing> _orders;
```

```
private List<Thing> _items;
```

Another method is the use of abstract class which is from Thing class. This specifies class don't have a body of itself but it takes it's form by the implementation of it derived class. Thing class Print() method, as the method work and we are able to hide details from both transaction and Batch by using the Thing abstract class. Abstract classes are only when important data is shown and what possible methods can it achieve as the body of the method can be found in it's derived classes instead

```
4 references
public abstract void Print();
```

```
public override void Print()
{
    Console.WriteLine($"Batch Sales: #{Number}, {Name}");

    decimal batchTotal = 0;

    if (_items.Count == 0)
    {
        Console.WriteLine("Empty Orders.");
    }
    else
    {
        foreach (var item in _items)
        {
            item.Print(); //print each line from items
            batchTotal += item.Total();
        }
        Console.WriteLine($"Total: ${batchTotal}");
    }

}
```

the body take forms of the it's derived classes

4.Can you think of a scenario or system design that resembles Task 1? Look at the classes and their interaction in Task 1 and identify a real-world system or approach that uses a similar relationship

-A scenario like Task1 would a checkout system in a bookstore. Because in checkout system it can process transaction, batch and calculate sales. It can calculate the total batch of books that we purchasing to generate a full price of mixture of books.