Sales.cs

```
using System;

namespace SemesterTest
{
    public class Sales
    {
        private List<Thing> _orders;

        public Sales()
        {
           _orders = new List<Thing> ();
        }

        public void Add(Thing thing)
        {
            _orders.Add(thing);
        }

        public void PrintOrders()
        {
            Console.WriteLine("Sales: ");
            Console.WriteLine();
            decimal total = 0;
            foreach (Thing order in _orders)
            {
                order.Print();
                Console.WriteLine("Total: " + "$" + order.Total());
                total += order.Total();
            }
            Console.WriteLine("Sales total: " + "$" + total.ToString());
        }
    }
}
```

Batch.cs

```
using System;

namespace SemesterTest
{
    public class Batch : Thing
    {
        private List<Thing> _items;

        public Batch(string number, string name) : base(number, name)
        {
            _items = new List<Thing>();
        }

        public void Add(Thing thing)
        {
            _items.Add(thing);
        }

        public override void Print()
```

```
        {
            Console.WriteLine("Batch sale: " + "#" + Number + ", " + Name);
            Console.WriteLine();
            if (_items.Count > 0 )
            {
                foreach (Thing item in _items)
                {
                    item.Print();
                }
            }
            else
            {
                Console.WriteLine("Empty order.");
            }
        }

        public override decimal Total()
        {
            decimal total = 0;
            foreach (Thing item in _items)
            {
                total += item.Total();
            }
            return total;
        }
    }

}
```

Thing.cs

```
using System;

namespace SemesterTest
{
    public abstract class Thing(string number, string name)
    {
        private string _number = number;
        private string _name = name;

        public abstract void Print();
        public abstract decimal Total();

        public string Number
        {
            get
            {
                return _number;
            }
        }
        public string Name
        {
            get
            {
                return _name;
            }
        }
    }
```

```
}
```

Transaction.cs

```
using System;

namespace SemesterTest
{
    public class Transaction : Thing
    {
        private decimal _amount;

        public Transaction(string number, string name, decimal amount) :
base(number, name)
        {
            _amount = amount;
        }

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

        public override decimal Total()
        {
            return _amount;
        }
    }
}
```

Program.cs

```
using System;

namespace SemesterTest
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            Sales sales = new Sales();

            Batch batch1 = new Batch("2024x01", "Computer Science Books");
            batch1.Add(new Transaction("1", "Introduction to Algorithm", 77.90m));
            batch1.Add(new Transaction("2", "The Pragmatic Programmer", 64.20m));
            batch1.Add(new Transaction("3", "Data Science for Business", 150.75m));

            Transaction transaction1 = new Transaction("01-01", "How The Steel Was
Tempered", 204.60m);
            Transaction transaction2 = new Transaction("03-01", "Gone with the
Wind", 70.30m);

            Batch batch2 = new Batch("2024x02", "Novels");
            batch2.Add(new Transaction("02-02", "The Lords of the Rings", 80m));
            batch2.Add(new Transaction("02-03", "A Song of Fire and Ice", 75m));
```

```
            Batch batch3 = new Batch("2024x012", "A Nested Batch");
            batch3.Add(batch1);
            batch3.Add(batch2);

            sales.Add(batch3);

            Batch batch4 = new Batch("2024x04", "Comics");

            sales.Add(transaction1);
            sales.Add(transaction2);
            sales.Add(batch4);

            sales.PrintOrders();
        }
    }
}
```