

Object Oriented Programming

Hurdle Task 1: Semester Test

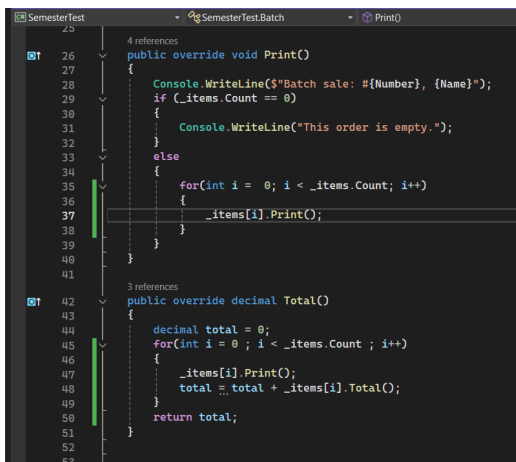
Name: Thenura Dulnath Kuruppuarachchi

ID: 103512993

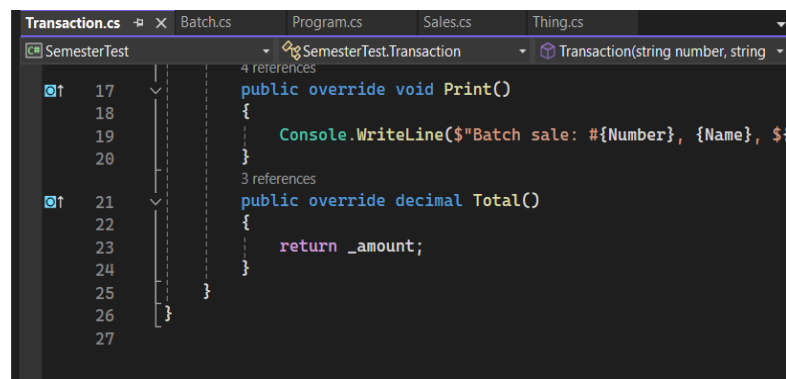
Task 2

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

The Principle of Polymorphism means the concept of accessing the different objects in different types through the same interface. If we get an example, a boy can be many things such as a student and a writer. So, in the task 1, Polymorphism can be seen used in the both Batch and Transaction classes.



```
26 public override void Print()
27 {
28     Console.WriteLine($"Batch sale: #{Number}, {Name}");
29     if (_items.Count == 0)
30     {
31         Console.WriteLine("This order is empty.");
32     }
33     else
34     {
35         for(int i = 0; i < _items.Count; i++)
36         {
37             _items[i].Print();
38         }
39     }
40 }
41
42 public override decimal Total()
43 {
44     decimal total = 0;
45     for(int i = 0; i < _items.Count; i++)
46     {
47         _items[i].Print();
48         total = total + _items[i].Total();
49     }
50     return total;
51 }
52
53
```



```
17 public override void Print()
18 {
19     Console.WriteLine($"Batch sale: #{Number}, {Name}, ${Total()};
20 }
21
22 public override decimal Total()
23 {
24     return _amount;
25 }
26
27
```

In here Both **Print ()** and **Total()** methods are being overridden in the **Batch** and **Transaction** classes and the **sales** class is using them without knowing their specific type.

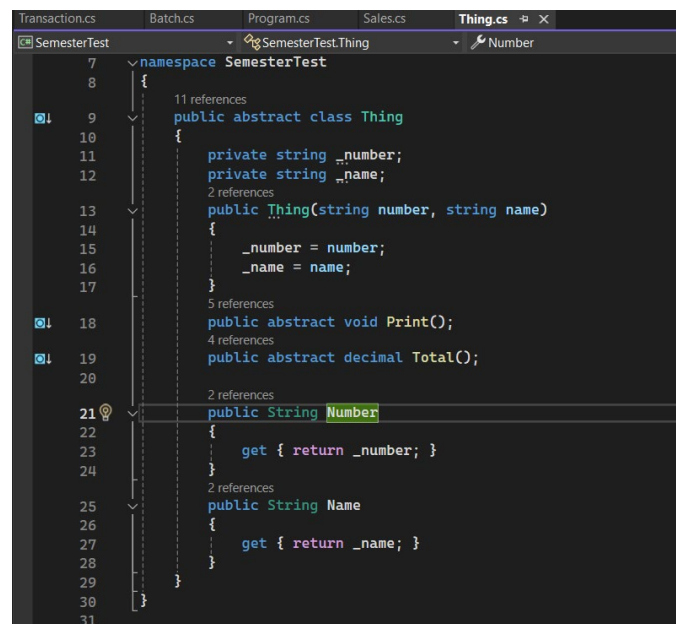
2. What is wrong with the class name *Thing*? Suggest a better name and explain the reasoning.

Using a class named as thing can be intimidating since it makes the purpose of it unclear and can mislead the other developers who will be working with this in the future. When suggesting a better name for this

class we can think of something related to order since this is a project based on the orders. So I can suggest replacing the name Thing with OrderEntry. Since it can give some kind of context to the what is the class for and what is the purpose of it. Also it can be very easy to handle the elements in the other classes as well such as Sales.

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

Abstraction method used to hide the unwanted parts of the project and show what only the user wanted so that the user can interact with them easily. If we give an example, in a mobile phone we can do many things like making a phone call, taking pictures, and playing games. But it doesn't show how they are being implemented in the system and how they work. So, in this project we can see it in the Thing class.



```
Transaction.cs Batch.cs Program.cs Sales.cs Thing.cs # X
SemesterTest SemesterTest.Thing Number
7 namespace SemesterTest
8 {
9     public abstract class Thing
10     {
11         private string _number;
12         private string _name;
13         public Thing(string number, string name)
14         {
15             _number = number;
16             _name = name;
17         }
18         public abstract void Print();
19         public abstract decimal Total();
20     }
21     public String Number
22     {
23         get { return _number; }
24     }
25     public String Name
26     {
27         get { return _name; }
28     }
29 }
30
31
```

In here it defines the common properties which are Number and Name and it also gives the common methods which are Print and Total for all types of orders, without providing specific implementations. This allows the Sales class to interact with its orders.

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.

In real-world, these types of implementations can be seen in the e-commerce platforms such as Amazon, Ebay which has systems that receiving and managing the orders. But when we get to the smaller versions such as this project we can see them in the online bookstore or and a small retail shops which are not selling the bulkier items. So, online smaller scale bookstores and the small retail shops are some of the shops that using this kind of systems.

```
1 using System;
2
3 namespace SemesterTest
4 {
5     public class Program
6     {
7         static void Main(string[] args)
8         {
9             Sales sales = new Sales();
10
11             //firstbatch
12             Batch FirstBatch = new Batch("2024x00001", "CompSci Books");
13             FirstBatch.Add(new Transaction("1", "Deep Learning in Python", 90));
14             FirstBatch.Add(new Transaction("2", "C# in Action", 80));
15             FirstBatch.Add(new Transaction("3", "Design Patterns", 40));
16             sales.Add(FirstBatch);
17
18             //ST
19             sales.Add(new Transaction("2024x00002", "HungerGames", 40));
20
21             //nest
22             Batch NestedBatch = new Batch("2024x00003", "Nested Batch");
23             NestedBatch.Add(new Transaction("NBT-ID-001", "NB Transaction 1", 40));
24             NestedBatch.Add(new Transaction("NBT-ID-002", "NB Transaction 2", 50));
25             sales.Add(NestedBatch);
26
27             //emp
28             Batch EmptyBatch = new Batch("2024x00004", "Fantasy Books");
29             sales.Add(EmptyBatch);
30
31             sales.PrintOrders();
32         }
33     }
34 }
35
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Linq;
5 using System.Numerics;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace SemesterTest
10 {
11     public class Batch : Thing
12     {
13         //private string _number;
14         private List<Thing> _items;
15
16         public Batch(string number, string name) : base(number, name)
17         {
18             _items = new List<Thing>();
19         }
20
21         public void Add(Thing thing)
22         {
23             _items.Add(thing);
24         }
25
26         public override void Print()
27         {
28             Console.WriteLine($"Batch sale: #{Number}, {Name}");
29             if (_items.Count == 0)
30             {
31                 Console.WriteLine("This order is empty.");
32             }
33             else
34             {
35                 for(int i = 0; i < _items.Count; i++)
36                 {
37                     _items[i].Print();
38                 }
39             }
40         }
41
42         public override decimal Total()
43         {
44             decimal total = 0;
45             for(int i = 0 ; i < _items.Count ; i++)
46             {
47                 _items[i].Print();
48                 total = total + _items[i].Total();
49             }
50         }
51     }
52 }
```

```
50         return total;
51     }
52 }
53 }
54
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SemesterTest
8 {
9
10     public class Sales
11     {
12         private List<Thing> _orders;
13         //private List<Transaction> _single_orders;
14
15         public Sales()
16         {
17             _orders = new List<Thing>();
18         }
19
20         public void Add(Thing thing)
21         {
22             _orders.Add(thing);
23         }
24
25         public void PrintOrders()
26         {
27             decimal TotSales = 0;
28             for (int i = 0; i < _orders.Count; i++)
29             {
30                 _orders[i].Print();
31                 TotSales = TotSales + _orders[i].Total();
32             }
33             Console.WriteLine($"Total Sales is: ${TotSales}");
34         }
35     }
36
37 }
38
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SemesterTest
8 {
9     public abstract class Thing
10    {
11        private string _number;
12        private string _name;
13        public Thing(string number, string name)
14        {
15            _number = number;
16            _name = name;
17        }
18        public abstract void Print();
19        public abstract decimal Total();
20
21        public String Number
22        {
23            get { return _number; }
24        }
25        public String Name
26        {
27            get { return _name; }
28        }
29    }
30 }
31
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SemesterTest
8 {
9     public class Transaction : Thing
10    {
11        private Decimal _amount;
12
13        public Transaction(String number, String name, Decimal amount):
14            base(number, name)
15        {
16            _amount = amount;
17        }
18        public override void Print()
19        {
20            Console.WriteLine($"Batch sale: #{Number}, {Name},
21                               ${_amount}");
22        }
23        public override decimal Total()
24        {
25            return _amount;
26        }
27    }
28 }
```