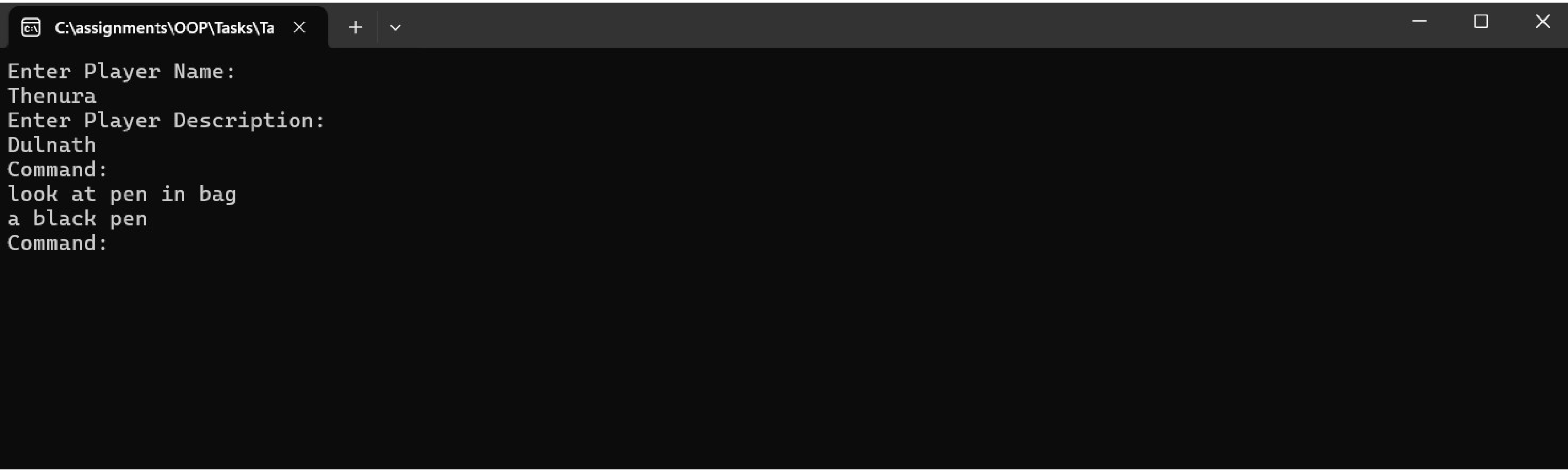

```
Enter Player Name:
Thenura
Enter Player Description:
Dulnath
Command:
look at pen in bag
a black pen
Command:
```

```
namespace Iteration5
{
    public class Program
    {
        static void Main(string[] args)
        {
            string name, description;
            Item item1, item2, item3;
            Bag bag0;
            LookCommand command = new LookCommand();

            item1 = new Item(new string[] { "computer" }, "a computer", "a small computer");
            item2 = new Item(new string[] { "bottle" }, "a bottle", "a white water bottle");
            item3 = new Item(new string[] { "pen" }, "a pen", "a black pen");

            bag0 = new Bag(new string[] { "bag" }, "a bag", "black color bag");

            Console.WriteLine("Enter Player Name: ");
            name = Console.ReadLine();

            Console.WriteLine("Enter Player Description: ");
            description = Console.ReadLine();

            Player Player1 = new Player(name, description);
            Player1.Inventory.Put(item1);
            Player1.Inventory.Put(item2);
            Player1.Inventory.Put(bag0);
            bag0.Inventory.Put(item3);

            while (true)
            {
                Console.WriteLine("Command: ");
                string userInput = Console.ReadLine();

                if (userInput == "stop")
                {
                    break;
                }
                else
                {
                    Console.WriteLine(command.Execute(Player1, userInput.Split()));
                }
            }
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class Bag : Item, IhaveInventory
    {
        private Inventory _inventory;
        public Bag(string[] ids, string name, string desc) : base(ids, name, desc)
        {
            _inventory = new Inventory();
        }
        public GameObject Locate(string id)
        {
            if (AreYou(id))
            {
                return this;
            }
            else
            {
                return _inventory.Fetch(id);
            }
        }
        public override string FullDescription
        {
            get
            {
                return $"In the {this.name} you can see:\n" + _inventory.ItemList;

            }
        }

        public Inventory Inventory
        {
            get { return _inventory; }
        }
    }
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public abstract class Command : IdentifiableObject
    {
        public Command(string[] ids) : base(ids)
        {

        }
        public abstract string Execute(Player p, string[] text);
    }
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class GameObject : IdentifiableObject
    {
        private string _description;
        private string _name;
        public GameObject(string[] ids, string name, string desc) : base(ids)
        {
            _description = desc;
            _name = name;
        }
        public string name
        {
            get { return _name; }
        }
        public string ShortDescription
        {
            get { return $"{_name} ({FirstID})"; }
        }
        public virtual string FullDescription
        {
            get { return _description; }
        }
    }

}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class IdentifiableObject
    {
        private List<string> _identifiers;

        public IdentifiableObject(string[] idents)
        {
            _identifiers = new List<string>();
            foreach (string ident in idents)
            {
                _identifiers.Add(ident.ToLower());
            }


        }

        public bool AreYou(string name)
        {
            foreach (string idents in _identifiers)
            {
                if (idents.ToLower() == name.ToLower())
                {
                    return true;
                }
            }

            return false;
        }

        public string FirstID
        {
            get
            {
                if (_identifiers.Count == 0)
                {
                    return "";
                }
                else
                {
                    return _identifiers.First();

                }
            }
        }

        public void AddIdentifier(string id)
        {
            _identifiers.Add(id.ToLower());
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public interface IhaveInventory
    {
        public string name
        {
            get;
        }
        public GameObject Locate(string id);

    }
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class Inventory
    {
        private List<Item> _items = new List<Item>();
        public Inventory()
        {

        }
        public bool HasItem(string id)
        {
            foreach (var item in _items)
            {
                if (item.AreYou(id))
                {
                    return true;
                }
            }
            return false;
        }
        public void Put(Item i)
        {
            _items.Add(i);
        }
        public Item Take(string id)
        {
            foreach (var item in _items)
            {
                if (item.AreYou(id))
                {
                    _items.Remove(item);
                    return item;
                }

            }
            return null;
        }
        public Item Fetch(string id)
        {
            foreach (var item in _items)
            {
                if (item.AreYou(id))
                {
                    return item;

                }
            }
            return null;
        }
        public string ItemList
        {
            get
            {
                string listItem = "";
                foreach (Item i in _items)
                {
                    listItem = listItem + i.ShortDescription + "\n";
                }
                return listItem;
            }
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class Item : GameObject
    {
        public Item(string[] idents, string name, string desc) : base(idents, name, desc)
        {

        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class LookCommand : Command
    {
        public LookCommand() : base(new string[] { "look" })
        {
        }

        public override string Execute(Player p, string[] text)
        {
            if ((text.Length != 3) && (text.Length != 5))
            {
                return "I don't know how to look like that";
            }
            else if (text[0] != "look")
            {
                return "Error in look input";
            }
            else if (text[1] != "at")
            {
                return "What do you want to look at?";
            }

            if ((text.Length == 5) && (text[3] != "in"))
            {
                return "What do you want to look in?";
            }

            String itemId = text[2];
            IhaveInventory container = p;

            if (text.Length == 5)
            {
                container = FetchContainer(p, text[4]);
                if (container == null)
                {
                    return $"I cannot find the {text[4]}";
                }
            }

            return LookAtIn(itemId, container);
        }

        private IhaveInventory FetchContainer(Player p, string containerId)
        {
            return p.Locate(containerId) as IhaveInventory;
        }

        private string LookAtIn(string thingId, IhaveInventory container)
        {
            var item = container.Locate(thingId);
            if (item != null)
            {
                return item.FullDescription;
            }
            else
            {
                return $"I can't find the {thingId}";
            }
        }
    }
}

```

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Iteration5
{
    public class Player : GameObject, IhaveInventory
    {
        private Inventory _inventory;
        public Player(string name, string desc) : base(new string[] { "me", "inventory" }, name, desc)
        {
            _inventory = new Inventory();
        }
        public GameObject Locate(string id)
        {
            if (AreYou(id))
            {
                return this;
            }
            else
            {
                return _inventory.Fetch(id);
            }
        }
        public override string FullDescription
        {
            get
            {
                return $"You are {this.name}. You are carrying:\n" + _inventory.ItemList;
            }
        }
        public Inventory Inventory
        {
            get
            {
                return _inventory;
            }
        }

    }
}

```