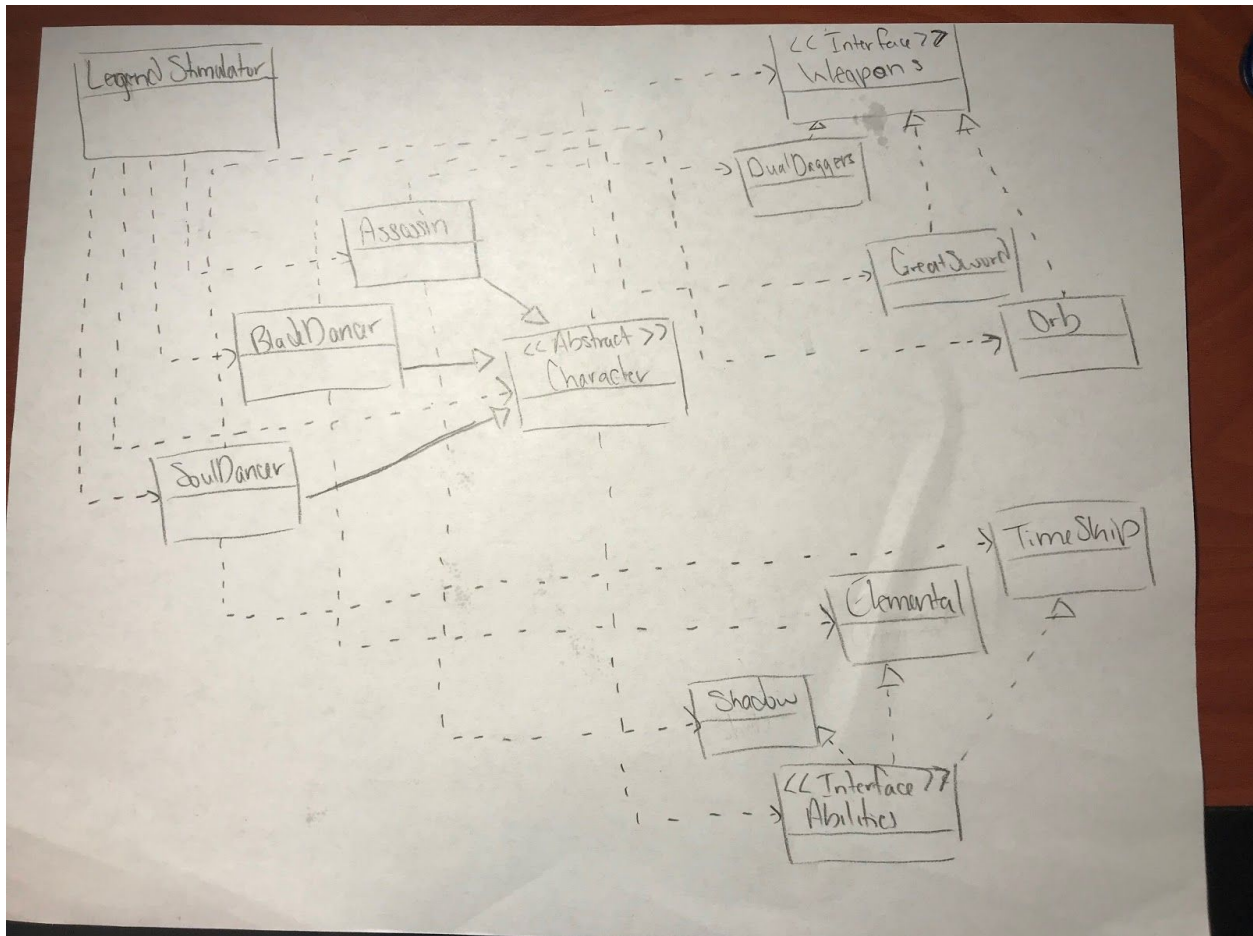


Name: Dominique Cooper
Eagle ID: 900929361
Course: CSCI 5335

UML Diagram:



Character Class:

```
public abstract class Character {  
    //Establish variables  
    Weapons Weapons;  
    Abilities Abilities;
```

```
    public Character() {  
    }  
}
```

```
    //Establish display method for character identification  
    abstract void display();
```

```
//Method for character to equipped weapon
public void weaponEquipped() {
    Weapons.weapons();
}
```

```
//Method for character to equipped ability
public void abilityEquipped() {
    Abilities.abilities();
}
}
```

Weapons interface and three behavior implementation classes

```
public interface Weapons {
    //Establish abstract weapon method
    public void weapons();
}
```

```
public class DualDaggers implements Weapons {
    //Establish method to choose the Dual Daggers as a weapon
    public void weapons() {
        System.out.println("Dual Daggers is my weapon of choice!");
    }
}
```

```
public class GreatSword implements Weapons {
    //Establish method to choose the Great Sword as a weapon
    public void weapons() {
        System.out.println("The Great Sword is what I wield!");
    }
}
```

```
public class Orb implements Weapons {
    //Establish method to choose the Orb as a weapon
    public void weapons() {
        System.out.println("Orb is equipped and ready!");
    }
}
```

```
}  
}
```

Abilities Interface and three behavior implementation classes:

```
public interface Abilities {  
    //Establish abstract ability method  
    public void abilities();  
}
```

```
public class Elemental implements Abilities {  
    //Establish method to choose Elemental powers as an ability  
    public void abilities() {  
        System.out.println("Elemental manipulation is what I can accomplish!");  
    }  
}
```

```
public class Shadow implements Abilities {  
    //Establish method to choose Shadow as an ability  
    public void abilities() {  
        System.out.println("I specialize in stealth and invisibility!");  
    }  
}
```

```
public class TimeSkip implements Abilities {  
    //Establish method to choose TimeSkip as an ability  
    public void abilities() {  
        System.out.println("I have the ability to manipulate time and space!");  
    }  
}
```

All Characters extended from the Character Class:

```
public class Assassin extends Character {  
    public Assassin() {
```

```

//Establish weapons and abilities for the Assassin class
Weapons = new DualDaggers();
Abilities = new Shadow();
}

//Method to display Assassin identification
public void display() {
    System.out.println("I'm a stealthy assassin!");
}
}

```

```

public class BladeDancer extends Character {
    public BladeDancer() {
        //Establish weapons and abilities for the BladeDancer class
        Weapons = new GreatSword();
        Abilities = new Elemental();
    }

    //Method to display BladeDancer identification
    public void display() {
        System.out.println("I am the mighty Blade Dancer!");
    }
}

```

```

public class SoulDancer extends Character{
    public SoulDancer() {
        //Establish weapons and abilities for the SoulDancer class
        Weapons = new Orb();
        Abilities = new TimeSkip();
    }

    //Method to display SoulDancer identification
    public void display( ) {
        System.out.println("I am the magical Soul Dancer!");
    }
}

```

Compile the test class, LegendStimulator.java:

```

public class LegendStimulator {
    public static void main(String[] args) {

```

```
Character Assassin = new Assassin();  
Assassin.display();  
Assassin.weaponEquipped();  
Assassin.abilityEquipped();
```

```
System.out.println();
```

```
Character BladeDancer = new BladeDancer();  
BladeDancer.display();  
BladeDancer.weaponEquipped();  
BladeDancer.abilityEquipped();
```

```
System.out.println();
```

```
Character SoulDancer = new SoulDancer();  
SoulDancer.display();  
SoulDancer.weaponEquipped();  
SoulDancer.abilityEquipped();
```

```
}  
}
```

Working Java Code:

