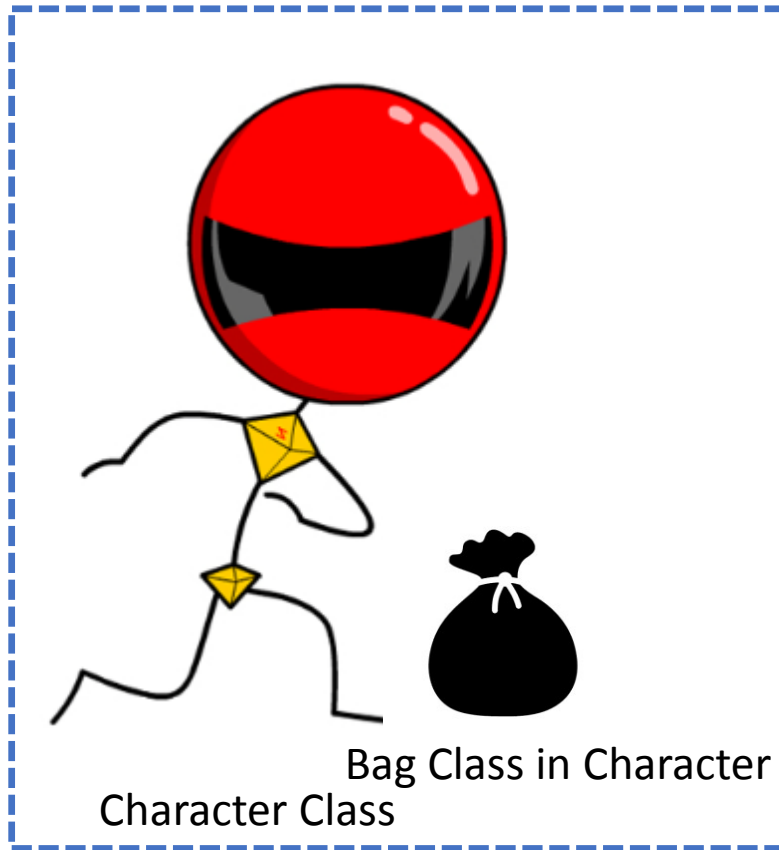


# Inner Class and Item\_Bag Class

Sungchul Lee

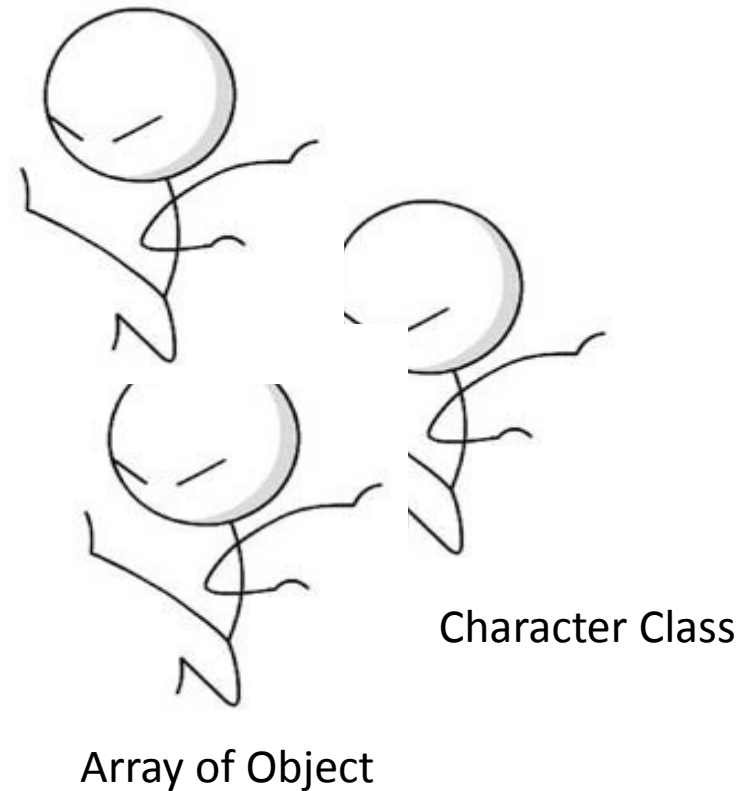
# Learning Object

Main Character



V.S.

Enemy



# Nested Class

- Classes that are defined inside another class
  - ❑ Like nested if statement, if statement inside other if
- Purpose of a nested class
  - ❑ Clearly group the nested class with its surrounding class, signaling that these **two classes are to be used together**
  - ❑ the nested class is only to be used from **inside its enclosing (owning) class.**
- Each character has own item bag
  - ❑ Use inner class for Bag class

Base Structure



```
class Character
{
    ...
    class Bag
    {
        ...
    }
}
```

# Inner Class

## ➤ Outer Class

- ❑ More than one inner class
- ❑ Can not use Inner class member in Outer's method

### ❖ Example 2

## ➤ Inner Class

- ❑ Can use outer's member
- ❖ Example 1
- ❑ Can not use other outer's member
- ❖ Need declare and assign new object
- ❑ Can not use **static keyword inside block**

```
class Outer
{
    statement 1;
    class Inner 1
    {
        statement 1-1;
    }
    class Inner 2
    {
        statement 2-1;
    }
}
```

# Bag Class (in Character Class)

- Hold items and money
- Inner Class
  - ❑ Outer: Character Class
- ***Instance Variable***
  - ❑ Weapon array, Armor Array, int money
- ***Methods***
  - ❑ Constructor for initializing bag for Weapon and Armor
  - ❑ Getter and setter for instance variable

# Practice

1. Open project (Starting from previous)
  - ☐ Project name: Game
2. Create a new Class in Character Class
  - ☐ Class name: Bag
3. Coding:
  - ☐ `import java.util.Random; // for Character, Weapon and Armor`
  - ☐ `import java.util.Scanner; // for Main class`
  - ☐ `import Equipment.*; // for Character`
  - ☐ `import Character.*; // for Main class`

# Practice – code (Bag)

```
100 public class Bag{
101     Weapon weaponArray[] = new Weapon[4];
102     Armor armorArray[] = new Armor[4];
103     int money = 120;
104     Bag(){
105         for(int i =0 ; i <4 ; i++) {
106             weaponArray[i] = new Weapon();
107             armorArray[i] = new Armor();
108         }
109     }
110     public Weapon[] getWeaponArray() {
111         return weaponArray;
112     }
113     public void setWeaponArray(Weapon[] weaponArray) {
114         this.weaponArray = weaponArray;
115     }
116     public Armor[] getArmorArray() {
117         return armorArray;
118     }
119     public void setArmorArray(Armor[] armorArray) {
120         this.armorArray = armorArray;
121     }
122     public int getMoney() {
123         return money;
124     }
125     public void setMoney(int money) {
126         this.money = money;
127     }
128 }
129 } // end of class
```

Constructor

Getter and Setter

Inside Character Class

# Practice – code (Character)

Character.java

```
1 package Character;
2 import java.util.Random;
4 public class Character {
5     private static String server_name = "CS172";
6     private Armor armor = new Armor();
7     private Weapon weapon = new Weapon();
8     private Bag bag = new Bag();
9     private int hp;
10    private int power;
11    private String name;
12    private int defense;
13    private int mp;
14    Random rnd = new Random();
15
16    public Character(String name) {
17        this.hp = rnd.nextInt(100);
18        this.power = rnd.nextInt(100);
19        this.name = name;
20        this.defense = rnd.nextInt(10);
21        this.mp = rnd.nextInt(100);
22    }
23    public Character(int hp, int power, String name) {
24        this.hp = hp;
25        this.power = power;
26        this.name = name;
27        this.defense = rnd.nextInt(10);
28        this.mp = rnd.nextInt(100);
29    }
```

Bag object

```
31 public Bag getBag() {
32     return bag;
33 }
34 public void setBag(Bag bag) {
35     this.bag = bag;
36 }
37 public Armor getArmor() {
38     return armor;
39 }
40 public void setArmor(Armor armor) {
41     this.armor = armor;
42 }
43 public Weapon getWeapon() {
44     return weapon;
45 }
46 public void setWeapon(Weapon weapon) {
47     this.weapon = weapon;
48 }
49 public static String getServer_name() {
50     return server_name;
51 }
52 public static void setServer_name(String server_name) {
53     Character.server_name = server_name;
54 }
55 public int getHp() {
56     return hp;
57 }
58 public void setHp(int hp) {
59     this.hp = hp;
60 }
61 public int getPower() {
62     return power;
63 }
```

Getter and Setter  
for bag



# Practice – code (Character) – cont.

```
64* public void setPower(int power) {
65     this.power = power;
66 }
67* public String getName() {
68     return name;
69 }
70 //no setter for name
71* public int getDefense() {
72     return defense;
73 }
74* public void setDefense(int defense) {
75     this.defense = defense;
76 }
77* public int getMp() {
78     return mp;
79 }
80* public void setMp(int mp) {
81     this.mp = mp;
82 }
83* public int hit_point(){
84     int hit_point = this.power+this.weapon.getPower();
85     return hit_point;
86 }
87* public int damage(int enemy_hit_point){
88     int damage = enemy_hit_point-this.defense - this.armor.getDefense();
89     if(damage<0){ // avoid healing by damage
90         damage =0;
91     }
92     this.hp=this.hp - damage;
93     if(this.hp<0){ // avoid minus hp
94         this.hp =0;
95     }
96     return damage;
97 }
```

Bag Class in  
Character Class

```
100* public class Bag{
101     Weapon weaponArray[] = new Weapon[4];
102     Armor armorArray[] = new Armor[4];
103     int money = 120;
104     Bag(){
105         for(int i =0 ; i <4 ; i++) {
106             weaponArray[i] = new Weapon();
107             armorArray[i] = new Armor();
108         }
109     }
110     public Weapon[] getWeaponArray() {
111         return weaponArray;
112     }
113     public void setWeaponArray(Weapon[] weaponArray) {
114         this.weaponArray = weaponArray;
115     }
116     public Armor[] getArmorArray() {
117         return armorArray;
118     }
119     public void setArmorArray(Armor[] armorArray) {
120         this.armorArray = armorArray;
121     }
122     public int getMoney() {
123         return money;
124     }
125     public void setMoney(int money) {
126         this.money = money;
127     }
128 }
129 } // end of class
```

# Practice – code (Main)

Import Character Package

```
*Main.java
1 import java.util.Scanner;
2 import Character.Character;
3 public class Main {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Scanner scanner = new Scanner(System.in);
7         System.out.println("Input Main Character Name:");
8         String main_name=scanner.nextLine();
9         System.out.println("Input Main Character Power:");
10        int main_power=scanner.nextInt();
11        System.out.println("Input Main Character HP:");
12        int main_hp=scanner.nextInt();
13
14        //Generate Main Character Object (Declare and initialize)
15        Character main_ch = new Character(main_hp,main_power,main_name);
16        show_status(main_ch);
17
18        Character enemies[] = new Character[5]; //Generate objects
19        for(int i =0 ; i<enemies.length;i++){
20            enemies[i]=new Character("enemy " + i); //initialize
21            show_status(enemies[i]);
22        }
23        show_status(enemies[0]);
24        enemies[0].damage(main_ch.hit_point()); //give damage to enemy
25        show_status(enemies[0]);
26
27        System.out.println("Main Character Equipment:");
28        System.out.println("Name:"+main_ch.getWeapon().getName());
29        System.out.println("Power:"+main_ch.getWeapon().getPower());
30        System.out.println("Name:"+main_ch.getArmor().getName());
31        System.out.println("Power:"+main_ch.getArmor().getDefense());
```

Give damage using  
damage method in  
Character class

Get Weapon  
and Armor info

# Practice – code (Main) – cont.

```
55     show_status(main_ch);
56     show_bag(main_ch);
57 }
58 public static void show_status(Character character) {
59     System.out.println("====Character Status =====");
60     System.out.println("Character Name: " + character.getName());
61     System.out.println("Character HP: " + character.getHp());
62     System.out.println("Character Power: " + character.getPower());
63     System.out.println("Character Defense: " + character.getDefense());
64     System.out.println("Character Mp: " + character.getMp());
65     System.out.println("Weapon name: " + character.getWeapon().getName());
66     System.out.println("Weapon Power: " + character.getWeapon().getPower());
67     System.out.println("Armor Name: " + character.getArmor().getName());
        System.out.println("Armor Defense: " + character.getArmor().getDefense());
    }
    public static void show_bag(Character character) {
        System.out.println("====Bag====");
        System.out.println("Money: " + character.getBag().getMoney());
        for(int i = 0; i < 4; i++) {
            System.out.println("Weapon Name: " + character.getBag().getWeaponArray()[i].getName());
            System.out.println("Weapon Power: " + character.getBag().getWeaponArray()[i].getPower());
            System.out.println("Armor Name: " + character.getBag().getArmorArray()[i].getName());
            System.out.println("Armor Defense: " + character.getBag().getArmorArray()[i].getDefense());
        }
    }
}
```

Check Bag

# Practice –Result

## ➤ Bag Class in Character Class

- ❑ int money

- ❑ Armor [4]

- ❑ Weapon [4]

- ❑ Getter and Setter

  - ❖ Array

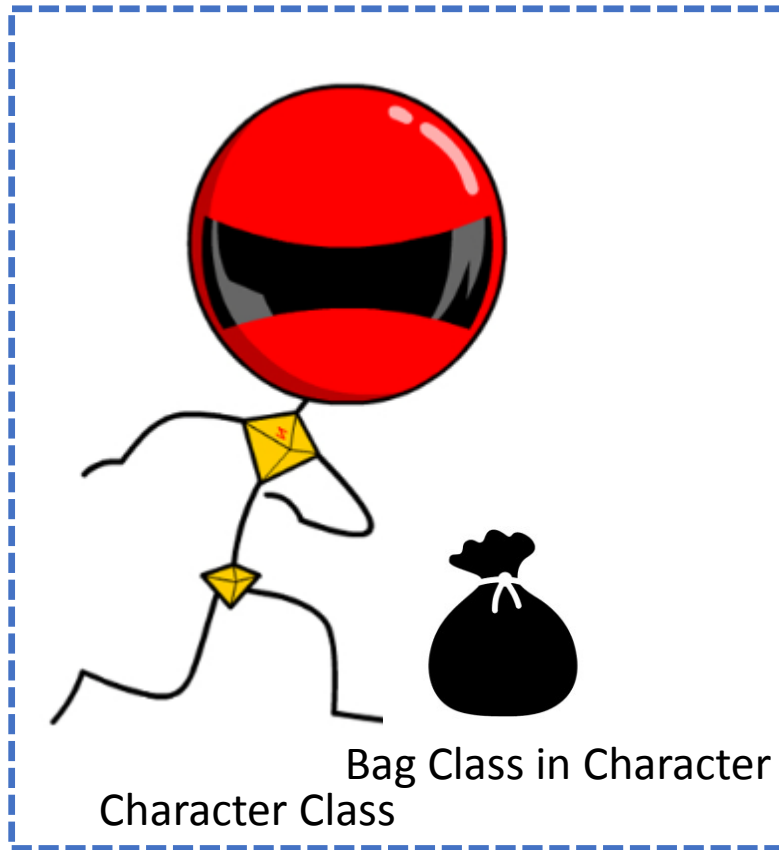


Result

```
Main Character Bag:
Name:Weapon46
Power:23
Name:Armor39
Defense:6
Name:Weapon21
Power:1
Name:Armor44
Defense:4
Name:Weapon39
Power:15
Name:Armor99
Defense:7
Name:Weapon39
Power:34
Name:Armor86
Defense:8
Money: 120
```

# Summary

Main Character



V.S.

Enemy

