

ASSIGNMENT NO 1

Student Name	Zarqa
Student ID	B2433131
Section	B
Task	Assignment 1 or Lab 01 or Lab 02 or Quiz 01 or Class Activity 01
Due Date	29-10-2024
Submitted On	29-10-2024
Submitted To	Sir Abdul Haseeb

QUESTION NO: 01

Q1) Imagine you are designing a basic software program to manage a zoo. In this task, you'll create classes and objects that represent different parts of a zoo, such as animals, staff members, and buildings. Each class should have attributes and methods relevant to its purpose.

EXPLANATION:

1. Animal Class:

- Represents animals in the zoo.
- Has attributes for species (type of animal) and age.
- showDetails() method displays the animal's species and age.

2. Staff Class:

- Represents zoo staff members.
- Has attributes for name, role, and shiftType.
- showDetails() method displays the staff member's details.

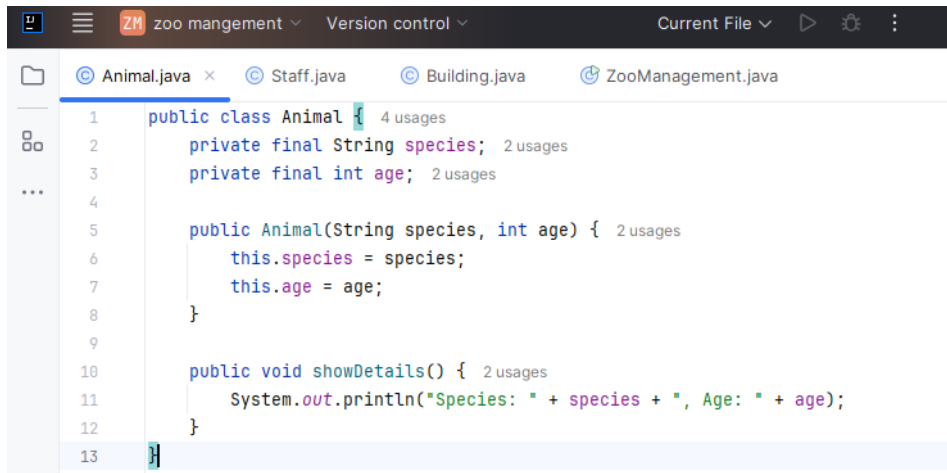
3. Building Class:

- Represents buildings in the zoo.
- Has attributes for name and purpose.
- showBuildingDetails() method displays the building's name and purpose.

4.ZooManagement Class:

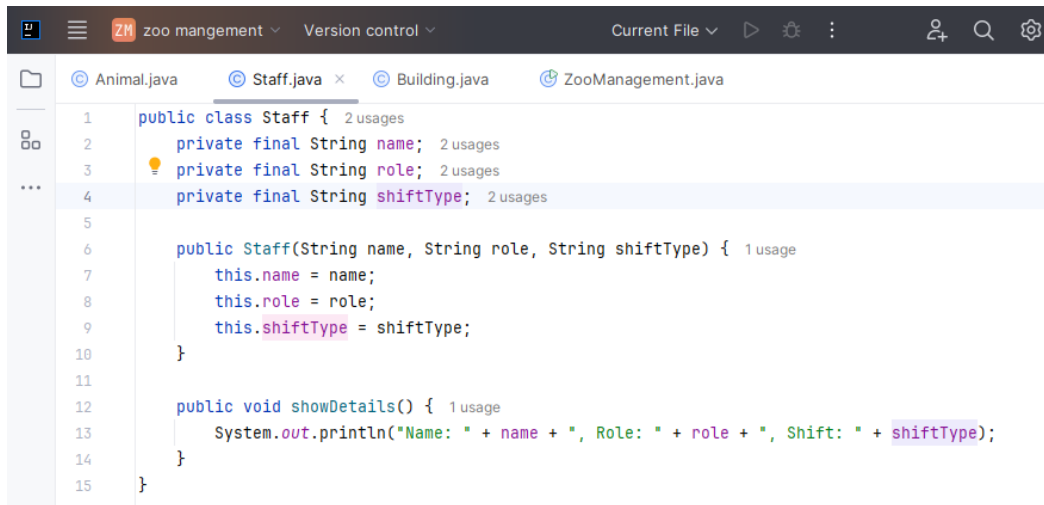
- The main class where the program runs.
- Creates objects of Animal, Staff, and Building, then displays their details.

- **ANIMAL.JAVA**



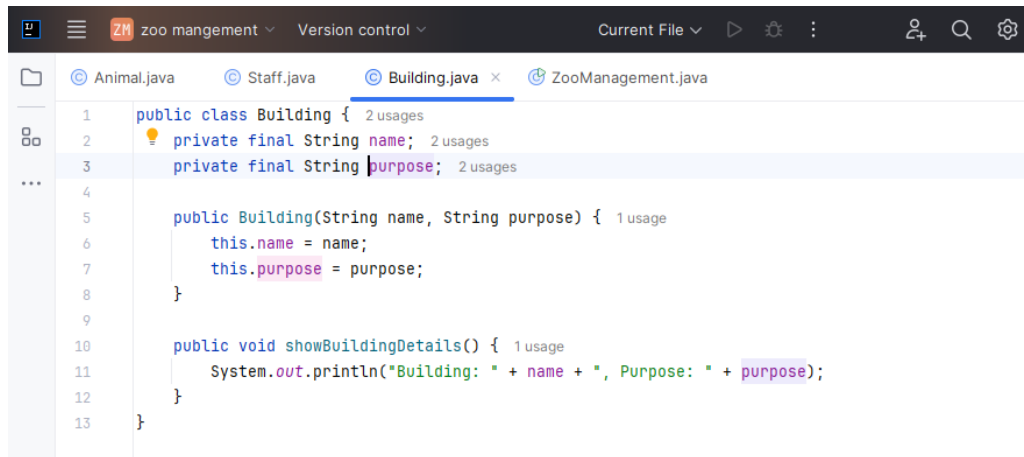
```
1 public class Animal { 4 usages
2     private final String species; 2 usages
3     private final int age; 2 usages
4
5     public Animal(String species, int age) { 2 usages
6         this.species = species;
7         this.age = age;
8     }
9
10    public void showDetails() { 2 usages
11        System.out.println("Species: " + species + ", Age: " + age);
12    }
13 }
```

- **STAFF.JAVA**



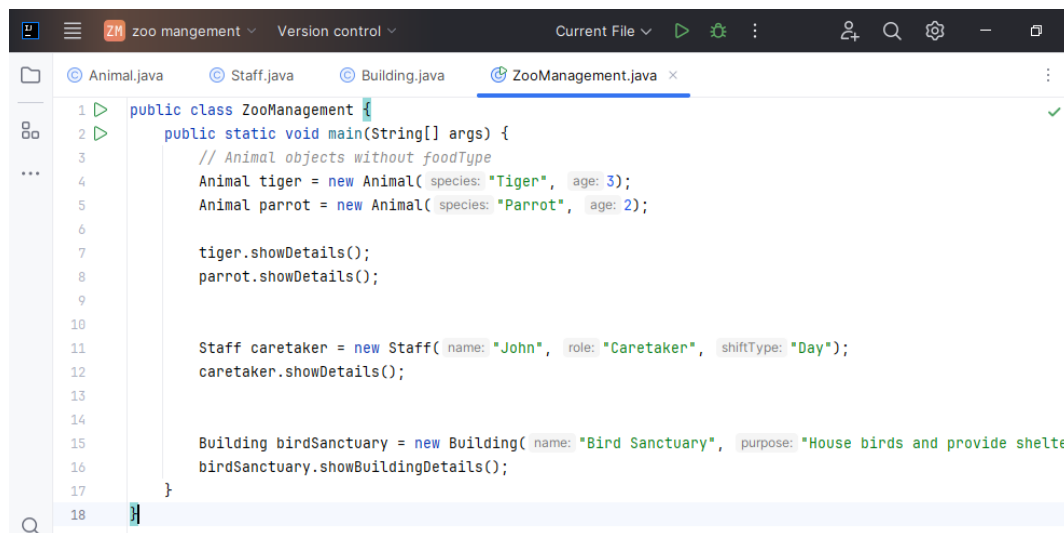
```
1 public class Staff { 2 usages
2     private final String name; 2 usages
3     private final String role; 2 usages
4     private final String shiftType; 2 usages
5
6     public Staff(String name, String role, String shiftType) { 1 usage
7         this.name = name;
8         this.role = role;
9         this.shiftType = shiftType;
10    }
11
12    public void showDetails() { 1 usage
13        System.out.println("Name: " + name + ", Role: " + role + ", Shift: " + shiftType);
14    }
15 }
```

• BUILDING.JAVA



```
1 public class Building { 2 usages
2     private final String name; 2 usages
3     private final String purpose; 2 usages
4
5     public Building(String name, String purpose) { 1 usage
6         this.name = name;
7         this.purpose = purpose;
8     }
9
10    public void showBuildingDetails() { 1 usage
11        System.out.println("Building: " + name + ", Purpose: " + purpose);
12    }
13 }
```

• ZOO MANAGEMENT.JAVA



```
1 public class ZooManagement {
2     public static void main(String[] args) {
3         // Animal objects without foodType
4         Animal tiger = new Animal( species: "Tiger", age: 3);
5         Animal parrot = new Animal( species: "Parrot", age: 2);
6
7         tiger.showDetails();
8         parrot.showDetails();
9
10
11        Staff caretaker = new Staff( name: "John", role: "Caretaker", shiftType: "Day");
12        caretaker.showDetails();
13
14
15        Building birdSanctuary = new Building( name: "Bird Sanctuary", purpose: "House birds and provide shelter");
16        birdSanctuary.showBuildingDetails();
17    }
18 }
```

• OUTPUT



```
Run ZooManagement x
"\"C:\\Program Files\\Java\\jdk-21\\bin\\java.exe\" \"-javaagent:E:\\IntelliJ IDEA Community Edition 2024.2.3\\lib\\idea_rt.jar\"
Species: Tiger, Age: 3
Species: Parrot, Age: 2
Name: John, Role: Caretaker, Shift: Day
Building: Bird Sanctuary, Purpose: House birds and provide shelter
Process finished with exit code 0
```

zoo magement > src > ZooManagement 18:2 CRLF UTF-8 4 spaces

QUESTION NO:02

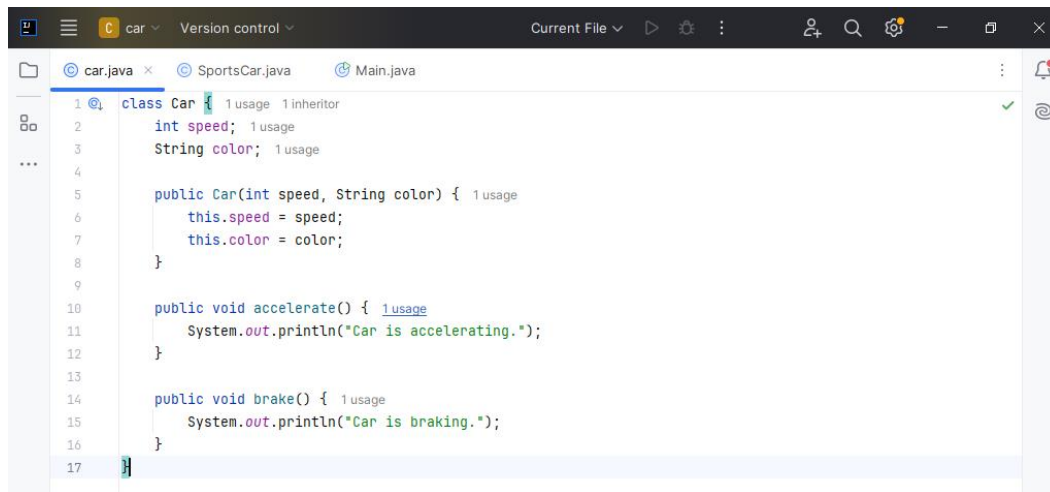
In this task, you'll explore inheritance by creating a SportsCar class based on an existing Car class. The Car class already has attributes like speed and color and methods like accelerate and brake.

Your job is to create a SportsCar class that inherits from the Car class and adds unique attributes or methods, such as turboBoost.

EXPLANTION:

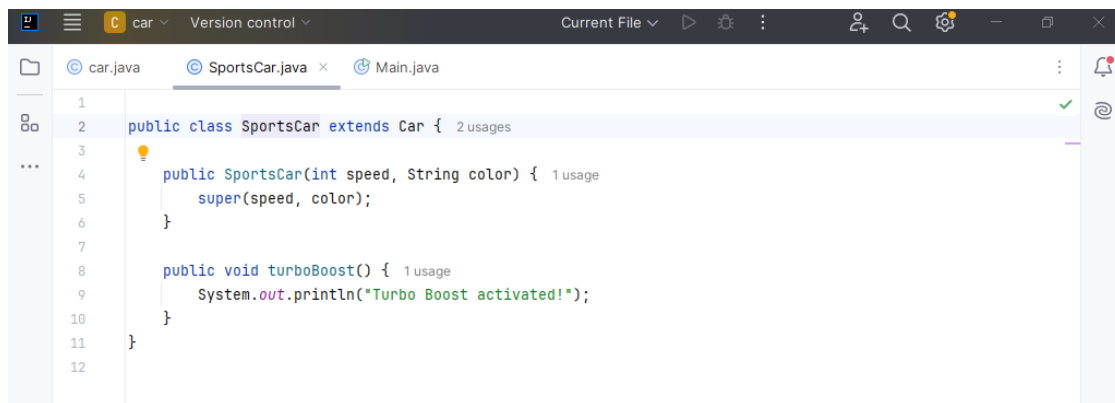
1. **Car Class:** This is the base class with two main attributes, speed and color, and two methods, accelerate() and brake(), which print simple messages to show the car is speeding up or slowing down.
2. The **SportsCar** inherits basic car features from Car and adds its own feature (turboBoost).
3. The **Main class** creates an instance of SportsCar and demonstrates how it can use both the inherited methods (accelerate, brake) and its own method (turboBoost).

- **CAR.JAVA:**



```
1 class Car { 1 usage 1 inheritor
2     int speed; 1 usage
3     String color; 1 usage
4
5     public Car(int speed, String color) { 1 usage
6         this.speed = speed;
7         this.color = color;
8     }
9
10    public void accelerate() { 1 usage
11        System.out.println("Car is accelerating.");
12    }
13
14    public void brake() { 1 usage
15        System.out.println("Car is braking.");
16    }
17 }
```

- **SPORTSCAR.JAVA:**



```
1
2 public class SportsCar extends Car { 2 usages
3
4     public SportsCar(int speed, String color) { 1 usage
5         super(speed, color);
6     }
7
8     public void turboBoost() { 1 usage
9         System.out.println("Turbo Boost activated!");
10    }
11 }
12 }
```

- **MAIN.JAVA:**



```
1 public class Main {
2     public static void main(String[] args) {
3
4         SportsCar mySportsCar = new SportsCar( speed: 100, color: "Red");
5
6
7         mySportsCar.accelerate();
8         mySportsCar.brake();
9         mySportsCar.turboBoost();
10    }
11 }
```

- **OUTPUT:**



The screenshot shows the 'Run' console in IntelliJ IDEA. The console title bar says 'Run' and 'Main x'. The output text is as follows:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:E:\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.
Car is accelerating.
Car is braking.
Turbo Boost activated!
Process finished with exit code 0
```

At the bottom of the console, the breadcrumb path is 'car > src > Main'. The status bar at the very bottom shows '11:2 LF UTF-8 4 spaces'.