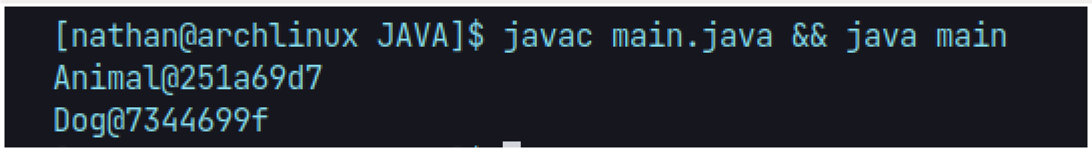
Create a class Animal with properties like name and age. Create a subclass Dog that inherits from Animal and adds a property breed. Demonstrate the use of constructors in both the Animal and Dog classes.

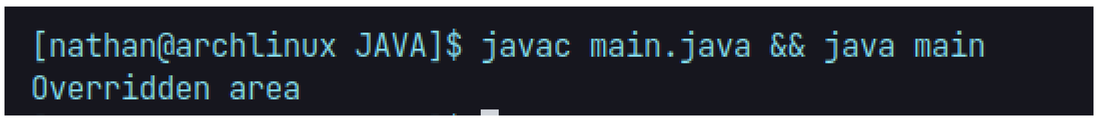
```java
class Animal{
    public String name;
    public int age;

    public Animal(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
class Dog extends Animal{
    public String breed;
    public Dog(String name, int age, String breed) {
        super(name, age); // For Calling The Parent
        this.breed = breed;
    }

}
public class main{
    public static void main(String[] args) {
        Animal animal = new Animal("Lion",5);
        Dog dog = new Dog("Woof",69,"Golden Retreiver");
        System.out.println(animal);
        System.out.println(dog);
    }
}
```



```
[nathan@archlinux JAVA]$ javac main.java && java main
Animal@251a69d7
Dog@7344699f
```

Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle

```java
class Shape{
    public void getPerimeter(){
        System.out.println("Shape's perimeter");
    }
    public void getArea(){
        System.out.println("Shape's area");
    }
}
class Circle extends Shape{
    public void getPerimeter(){
        System.out.println("Overridden perimeter");
    }
    public void getArea(){
        System.out.println("Overridden area");
    }

}
public class main{
    public static void main(String[] args) {
        Shape circle = new Circle();
        circle.getArea();
    }
}
```



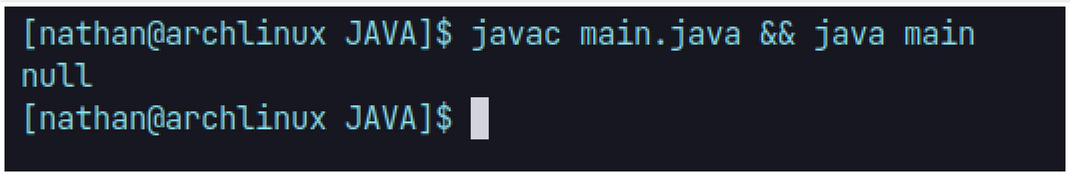
```
[nathan@archlinux JAVA]$ javac main.java && java main
Overridden area
```

Extend the Animal and Dog example by adding a constructor to the Animal class that takes a name parameter. In the Dog class, use the super keyword to call the constructor of the Animal class. Create instances of Dog and demonstrate the use of the super keyword.

```
class Animal{
    public String name;
    public int age;

    public Animal(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
class Dog extends Animal{
    public String breed;
    public Dog(String name, int age, String breed) {
        super(name, age); // For Calling The Parent
        this.breed = breed;
    }

}
public class main{
    public static void main(String[] args) {
        Animal animal = new Animal("Lion",5);
        Dog dog = new Dog("Woof",69,"Golden Retreiver");
        System.out.println(animal);
        System.out.println(dog);
    }
}
```

Create a class Person with a protected attribute **address**. Extend it with a subclass Employee that adds a **department** attribute. Demonstrate how the protected keyword allows access to the address property in the Employee subclass.

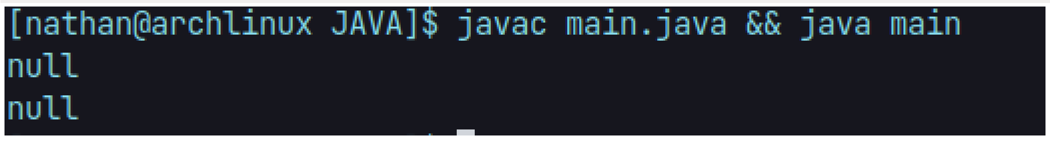
```
class Person{
    protected String address;
}
class Employee extends Person{
    public String department;
}
public class main{
    public static void main(String[] args) {
        Employee employee = new Employee();
        System.out.println(employee.address);
    }
}
```



```
[nathan@archlinux JAVA]$ javac main.java && java main
null
[nathan@archlinux JAVA]$ 
```

Create class Parent with a private variable, a protected variable, and a public variable. Create a subclass Child and demonstrate how each type of variable is accessed (or not accessed) within the subclass.

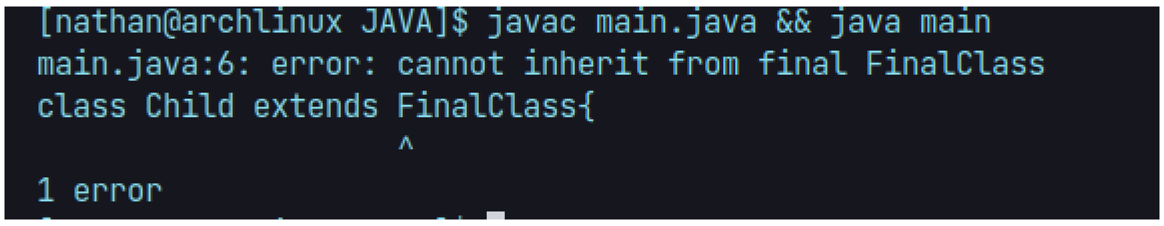
```
class Parent{
    protected String a;
    private String b;
    public String c;
}
class Child extends Parent{
}
public class main{
    public static void main(String[] args) {
        Child child = new Child();
        System.out.println(child.a);
        System.out.println(child.c);
        System.out.println(child.b);  // Errors Out As It Is Private
    }
}
```



```
[nathan@archlinux JAVA]$ javac main.java && java main
null
null
```

Create a final class FinalClass. Attempt to extend it with another class and observe the compiler error. Also, create a final method within a class and try to override it in a subclass.

```
final class FinalClass{
    public void display() {
        System.out.println("This is a final class.");
    }
}
class Child extends FinalClass{
    public void display() {
        System.out.println("This is an attempt to override final class.");
    }
}
public class main{
    public static void main(String[] args) {
    }
}
```



```
[nathan@archlinux JAVA]$ javac main.java && java main
main.java:6: error: cannot inherit from final FinalClass
class Child extends FinalClass{
                    ^
1 error
```

```
final class Calculator{
    public void add(int ...numbers){
        int sum = 0;
        for (int number: numbers){
            sum += number;
        }
        System.out.println(sum);
    }
    public void add(double ...numbers){
        double sum = 0;
        for (double number: numbers){
            sum += number;
        }
        System.out.println(sum);
    }
}
public class main{
    public static void main(String[] args) {
        Calculator calculator = new Calculator();
        calculator.add(1,2);
        calculator.add(1,2,3);
        calculator.add(1.1,2.2);
        calculator.add(1.1,2.2,3.3);
    }
}
```

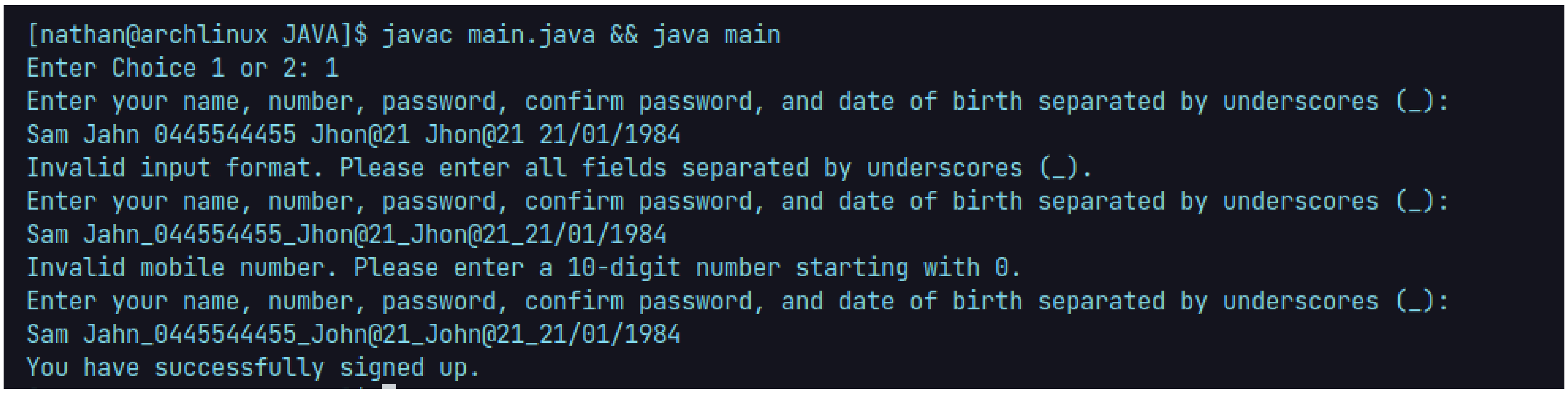
Section -2

Case Study

```
import java.util.Scanner;

public class main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Choice 1 or 2: ");
        int userChoice = Integer.parseInt(scanner.nextLine());

        if (userChoice == 1) {
            while (true) {
                System.out.println("Enter your name, number, password,
confirm password, and date of birth separated by underscores (_): ");
                String userInput = scanner.nextLine();
                String[] inputs = userInput.split("_");

                if (inputs.length != 5) {
                    System.out.println("Invalid input format. Please enter all
fields separated by underscores (_).");
```

```
                    continue;
                }

                if (inputs[0].length() <= 4) {
                    System.out.println("Invalid full name. Please enter a name
longer than four characters.");
                    continue;
                }

                if (!inputs[1].matches("^0\\d{9}$")) {
                    System.out.println("Invalid mobile number. Please enter a
10-digit number starting with 0.");
                    continue;
                }

                if (!inputs[2].matches("^[A-Za-z]+[@&]\\d+$")) {
                    System.out.println("Invalid password format. Password must
start with letters, contain '@' or '&', and end with digits.");
                    continue;
                }

                if (!inputs[2].equals(inputs[3])) {
                    System.out.println("Passwords do not match. Please enter
matching passwords.");
                    continue;
                }

                String[] dateParts = inputs[4].split("/");
                if (dateParts.length != 3) {
                    System.out.println("Invalid date of birth format. Please
enter DD/MM/YYYY or MM/DD/YYYY.");
                    continue;
                }

                int birthYear = Integer.parseInt(dateParts[2]);
                int age = 2024 - birthYear;

                if (age < 21) {
                    System.out.println("You must be at least 21 years old.");
                    continue;
                }

                System.out.println("You have successfully signed up.");
                break;
            }
```

```
        } else {
            System.out.println("Goodbye!");
        }
    }
}
```



```
[nathan@archlinux JAVA]$ javac main.java && java main
Enter Choice 1 or 2: 1
Enter your name, number, password, confirm password, and date of birth separated by underscores (_):
Sam Jahn 0445544455 Jhon@21 Jhon@21 21/01/1984
Invalid input format. Please enter all fields separated by underscores (_).
Enter your name, number, password, confirm password, and date of birth separated by underscores (_):
Sam Jahn_044554455_Jhon@21_Jhon@21_21/01/1984
Invalid mobile number. Please enter a 10-digit number starting with 0.
Enter your name, number, password, confirm password, and date of birth separated by underscores (_):
Sam Jahn_0445544455_John@21_John@21_21/01/1984
You have successfully signed up.
```