

```
import java.util.Scanner;
```

```
class WrongAgeException extends Exception {  
    public WrongAgeException(String message) {  
        super(message);  
    }  
}
```

```
class SonAgeException extends Exception {  
    public SonAgeException(String message) {  
        super(message);  
    }  
}
```

```
class Father {  
    private int age;  
  
    public Father(int age) throws WrongAgeException {  
        if (age < 0) {  
            throw new WrongAgeException("Wrong age");  
        }  
        this.age = age;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

```
class Son extends Father {  
    private int sonAge;
```

```

public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
    super(fatherAge);
    if (sonAge >= fatherAge) {
        throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
    }
    this.sonAge = sonAge;
}

public int getSonAge() {
    return sonAge;
}
}

public class AppaMaga {
    public static void main(String[] args) {
        System.out.println("name: AARUSHA\n USN: 1BM23CS005");

        Scanner sc = new Scanner(System.in); // Scanner should be outside the loop to avoid creating
        multiple instances

        while (true) {
            try {
                // Get Father's Age
                System.out.print("Enter Father's Age: ");
                int fatherAge = getValidAge(sc);

                // Get Son's Age
                System.out.print("Enter Son's Age: ");
                int sonAge = getValidAge(sc);
            }
        }
    }
}

```

```

        // Create the Son object
        Son son = new Son(fatherAge, sonAge);

        System.out.println("Accepted Successfully");

    } catch (WrongAgeException | SonAgeException e) {

        System.out.println(e.getMessage());

    }

    // Ask if the user wants to continue
    System.out.println("Would you like to re-enter details (Y/n)");

    String input = sc.next();

    if (input.equalsIgnoreCase("n")) {

        break;

    }

}

sc.close(); // Close the scanner outside the loop
}

// Helper method to get valid age input
private static int getValidAge(Scanner sc) {

    while (true) {

        if (sc.hasNextInt()) {

            int age = sc.nextInt();

            if (age >= 0) {

                return age;

            } else {

                System.out.print("Age must be a non-negative integer. Try again: ");

            }

        } else {

            sc.next(); // Clear the invalid input

```

```
        System.out.print("Invalid input. Please enter a valid age: ");  
    }  
}  
}  
}
```

Output:

```
name: AARUSHA  
USN: 1BM23CS005  
Enter Father's Age: 42  
Enter Son's Age: 18  
Accepted Succesfully  
Would you like to re-enter details (Y/n)  
y  
Enter Father's Age: 55  
Enter Son's Age: 58  
Son's age cannot be greater than or equal to father's age  
Would you like to re-enter details (Y/n)  
n
```