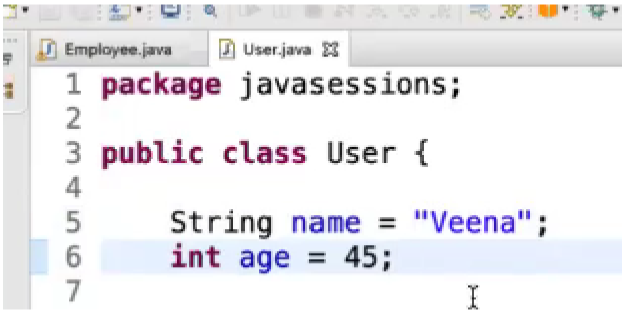
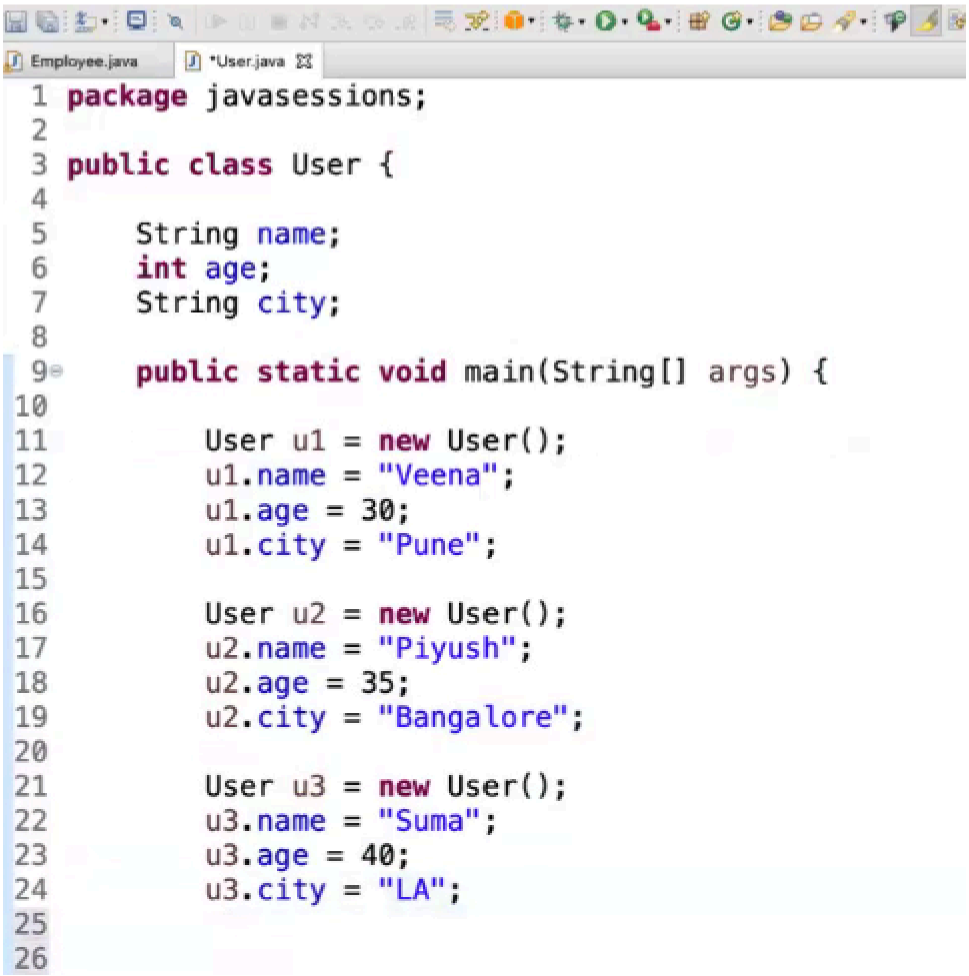
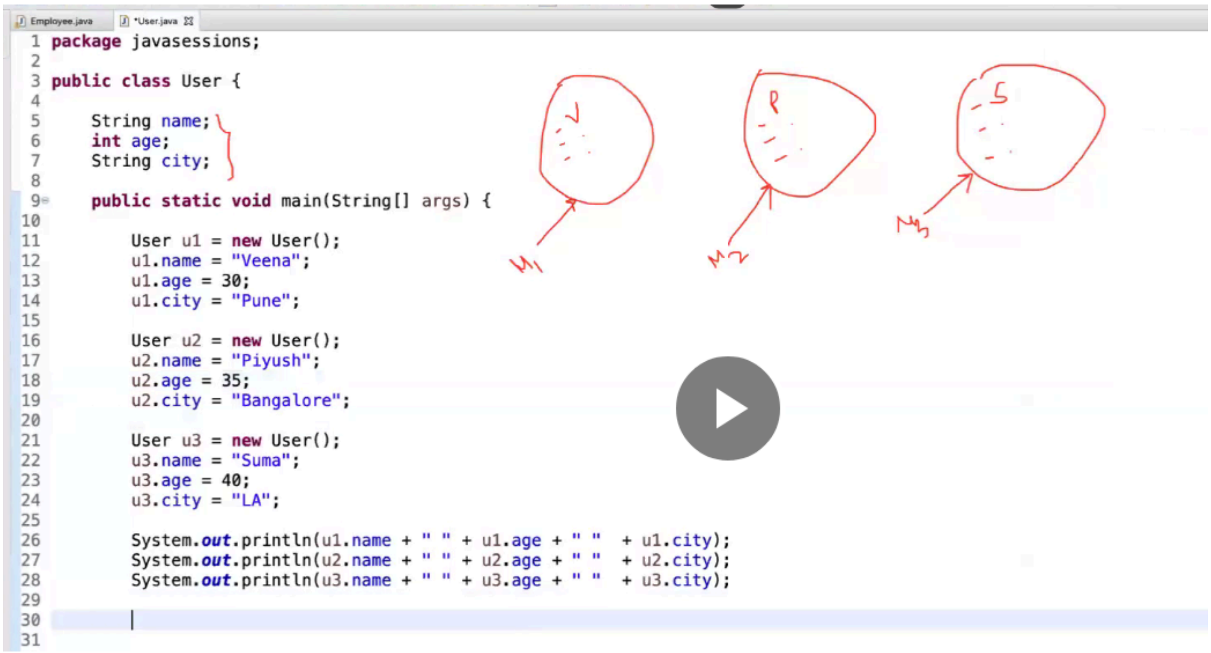
Not good practice-



```
package javasessions;

public class User {

    String name = "Veena";
    int age = 45;

```



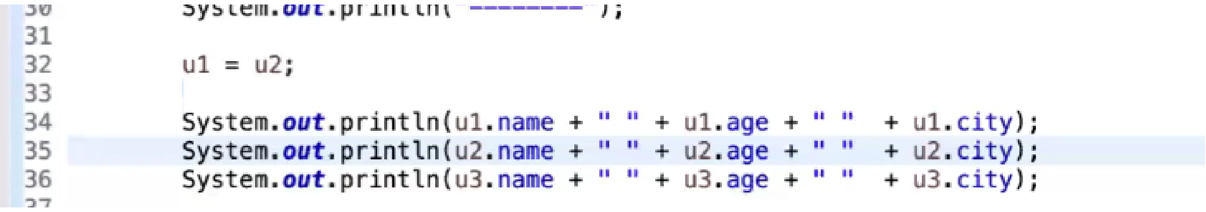
```
package javasessions;

public class User {

    String name;
    int age;
    String city;

    public static void main(String[] args) {

        User u1 = new User();
        u1.name = "Veena";
        u1.age = 30;
        u1.city = "Pune";

        User u2 = new User();
        u2.name = "Piyush";
        u2.age = 35;
        u2.city = "Bangalore";

        User u3 = new User();
        u3.name = "Suma";
        u3.age = 40;
        u3.city = "LA";

```



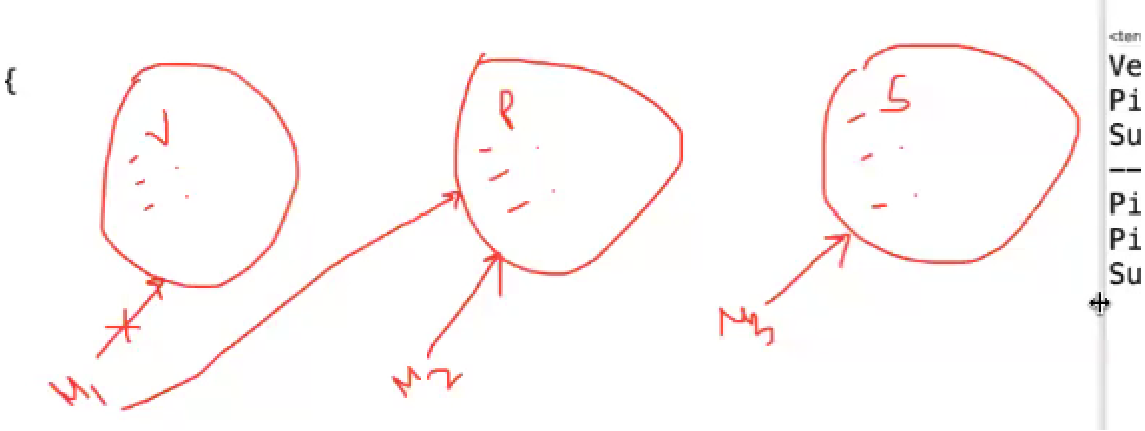
```java
package javasessions;

public class User {

    String name;
    int age;
    String city;

    public static void main(String[] args) {

        User u1 = new User();
        u1.name = "Veena";
        u1.age = 30;
        u1.city = "Pune";

        User u2 = new User();
        u2.name = "Piyush";
        u2.age = 35;
        u2.city = "Bangalore";

        User u3 = new User();
        u3.name = "Suma";
        u3.age = 40;
        u3.city = "LA";

        System.out.println(u1.name + " " + u1.age + " "  + u1.city);
        System.out.println(u2.name + " " + u2.age + " "  + u2.city);
        System.out.println(u3.name + " " + u3.age + " "  + u3.city);

        
```

Veena 30 pune

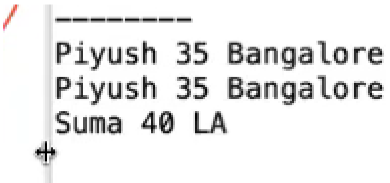
Piyush 35 bangalore and so on.



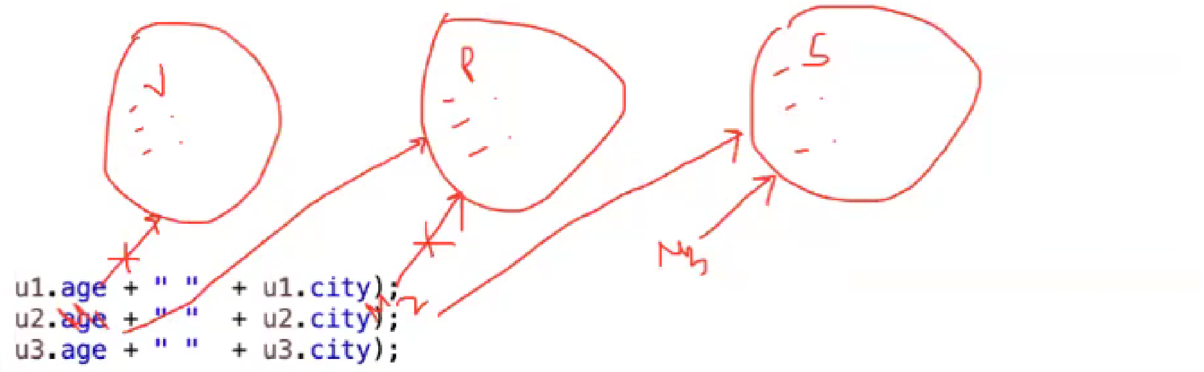
```java
        System.out.println("---------");

        u1 = u2;

        System.out.println(u1.name + " " + u1.age + " "  + u1.city);
        System.out.println(u2.name + " " + u2.age + " "  + u2.city);
        System.out.println(u3.name + " " + u3.age + " "  + u3.city);
```



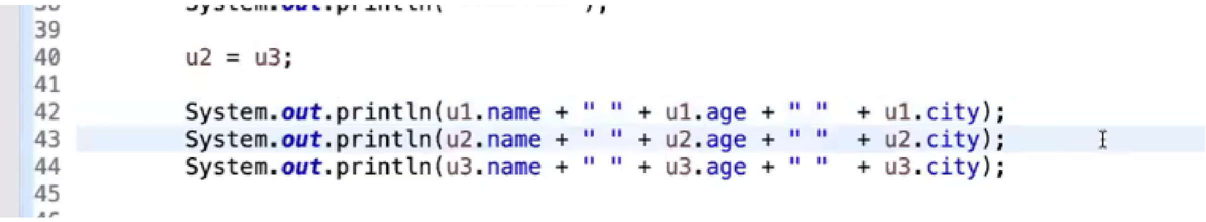
```
<term
Ve
Pi
Su
--
Pi
Pi
Su
```



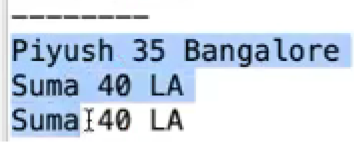
```
---------
Piyush 35 Bangalore
Piyush 35 Bangalore
Suma 40 LA
```



```
u1.age + " "  + u1.city);
u2.age + " "  + u2.city);
u3.age + " "  + u3.city);
```

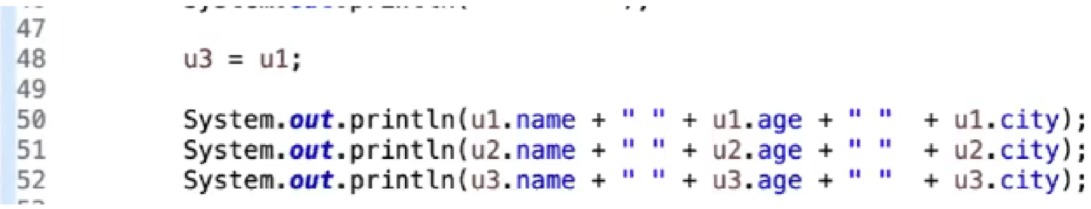


```
        u2 = u3;

        System.out.println(u1.name + " " + u1.age + " "  + u1.city);
        System.out.println(u2.name + " " + u2.age + " "  + u2.city);
        System.out.println(u3.name + " " + u3.age + " "  + u3.city);

```

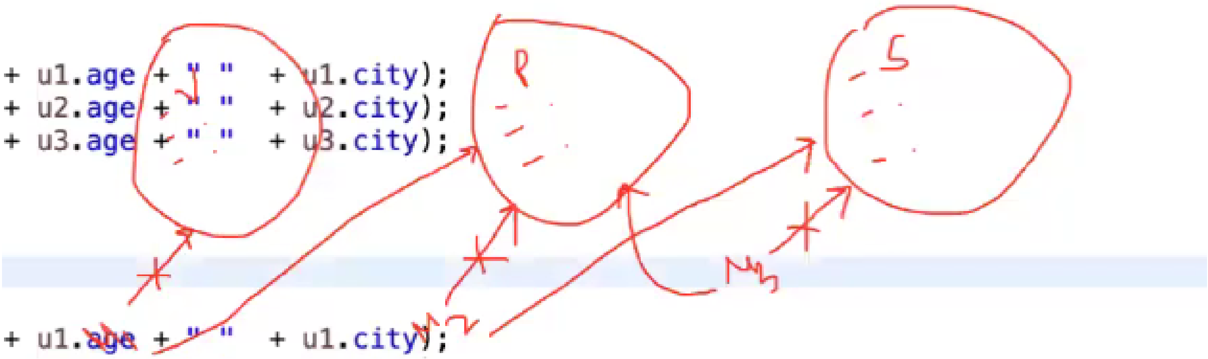


```
---------
Piyush 35 Bangalore
Suma 40 LA
Suma 40 LA
```

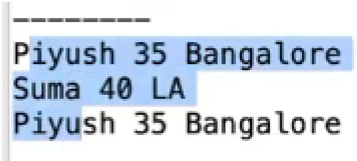
In java one object can have multiple references as seen above.



```
        u3 = u1;

        System.out.println(u1.name + " " + u1.age + " "  + u1.city);
        System.out.println(u2.name + " " + u2.age + " "  + u2.city);
        System.out.println(u3.name + " " + u3.age + " "  + u3.city);
```

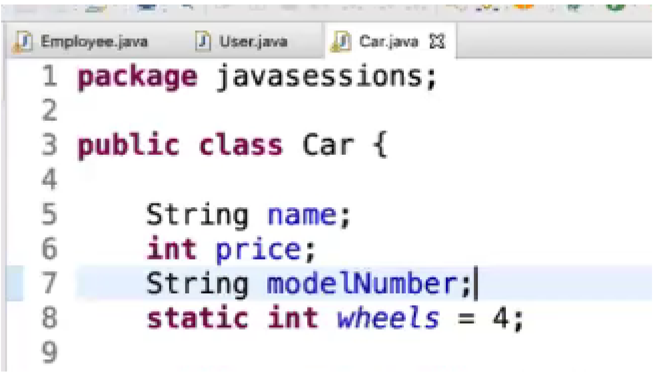


```
+ u1.age + " "  + u1.city);
+ u2.age + " "  + u2.city);
+ u3.age + " "  + u3.city);

+ u1.age + " "  + u1.city);
```

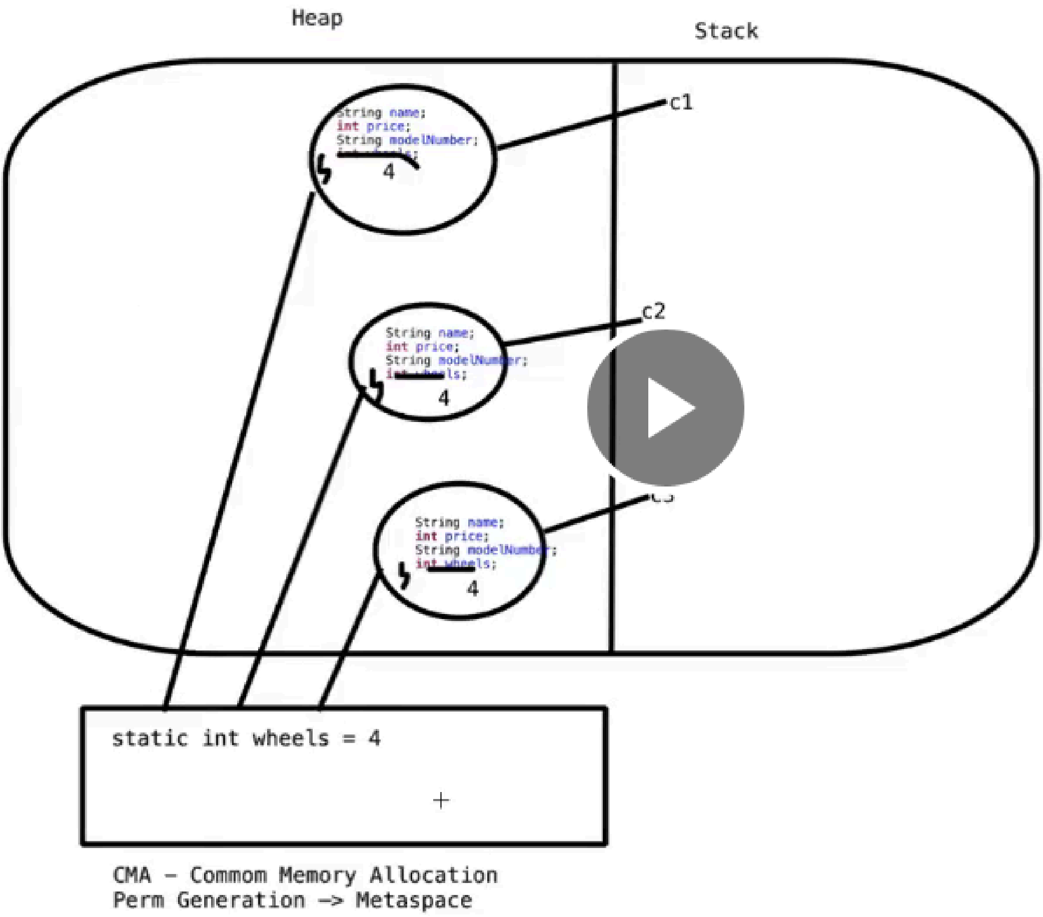


```
---------
Piyush 35 Bangalore
Suma 40 LA
Piyush 35 Bangalore
```

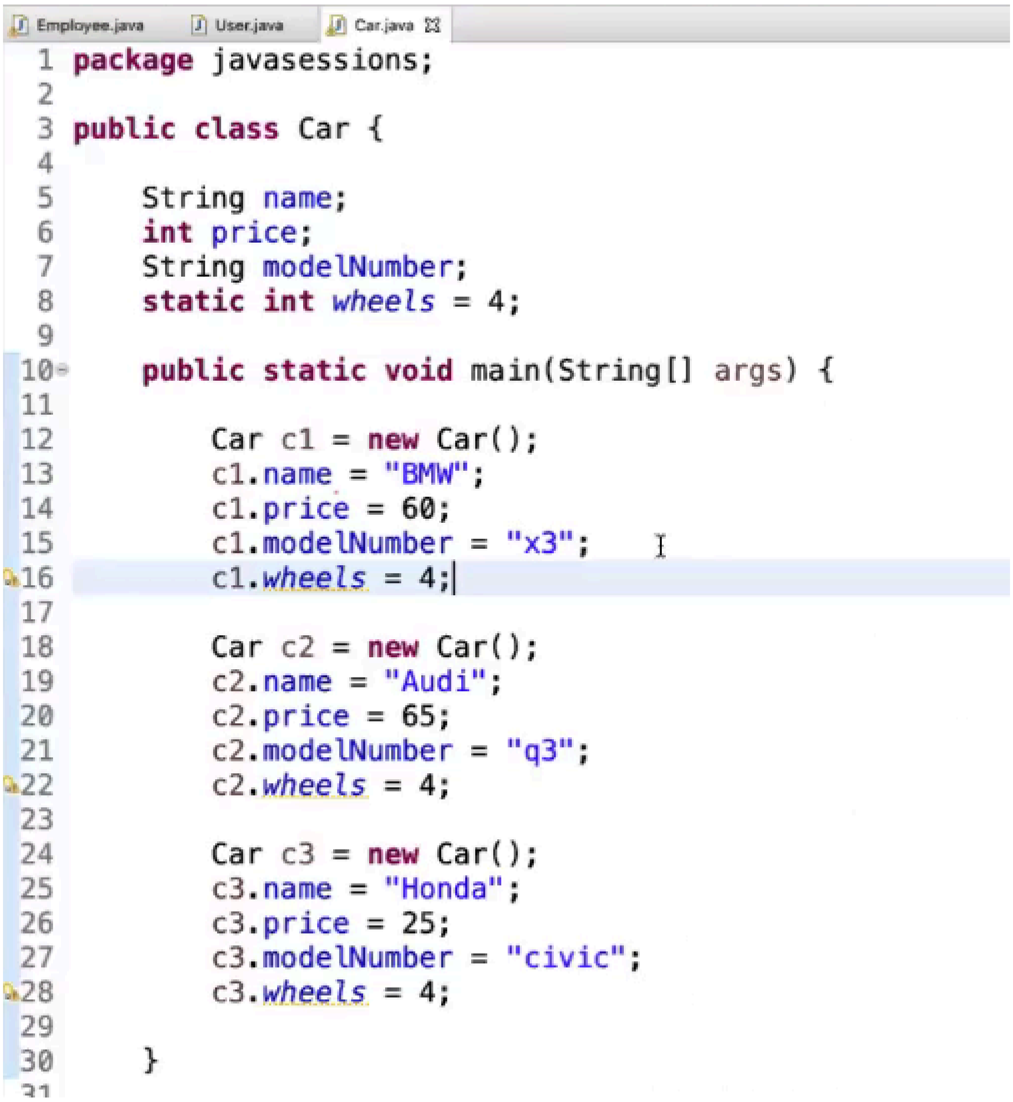
Static variables-



```java
package javasessions;

public class Car {

    String name;
    int price;
    String modelNumber;
    static int wheels = 4;

```

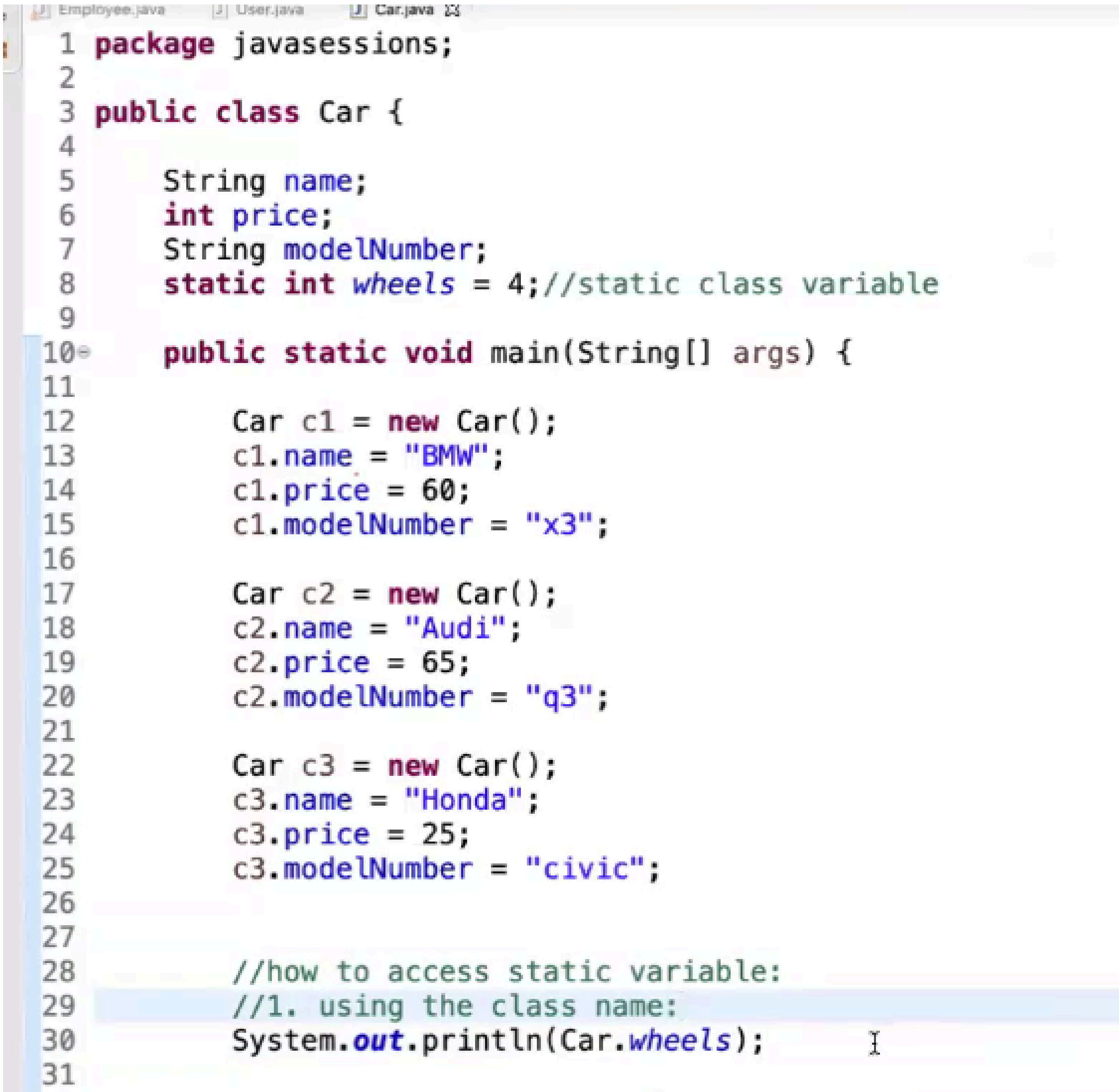


Warning –



```java
package javasessions;

public class Car {

    String name;
    int price;
    String modelNumber;
    static int wheels = 4;

    public static void main(String[] args) {

        Car c1 = new Car();
        c1.name = "BMW";
        c1.price = 60;
        c1.modelNumber = "x3";
        c1.wheels = 4;

        Car c2 = new Car();
        c2.name = "Audi";
        c2.price = 65;
        c2.modelNumber = "q3";
        c2.wheels = 4;

        Car c3 = new Car();
        c3.name = "Honda";
        c3.price = 25;
        c3.modelNumber = "civic";
        c3.wheels = 4;

    }
```

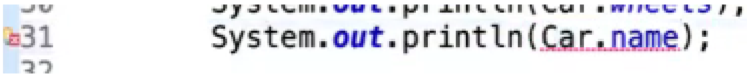
Static variables should be accessed in static way.



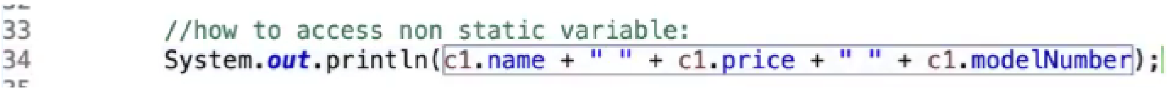
```java
package javasessions;

public class Car {

    String name;
    int price;
    String modelNumber;
    static int wheels = 4;//static class variable

    public static void main(String[] args) {

        Car c1 = new Car();
        c1.name = "BMW";
        c1.price = 60;
        c1.modelNumber = "x3";

        Car c2 = new Car();
        c2.name = "Audi";
        c2.price = 65;
        c2.modelNumber = "q3";

        Car c3 = new Car();
        c3.name = "Honda";
        c3.price = 25;
        c3.modelNumber = "civic";


        //how to access static variable:
        //1. using the class name:
        System.out.println(Car.wheels);

```

Error-

//Cannot make a static reference to the non-static field car2.name

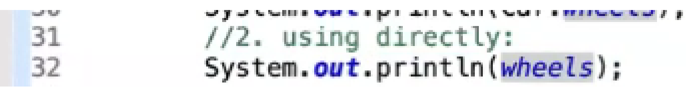


```java
        System.out.println(Car.wheels);
        System.out.println(Car.name);
```



```java
        //how to access non static variable:
        System.out.println(c1.name + " " + c1.price + " " + c1.modelNumber);
```

Bmw 60 x3

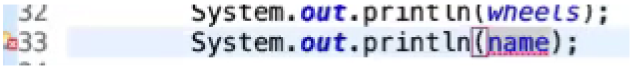


```java
        System.out.println(Car.wheels);
        //2. using directly:
        System.out.println(wheels);
```

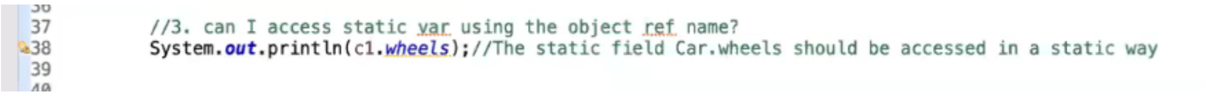
4

Error-

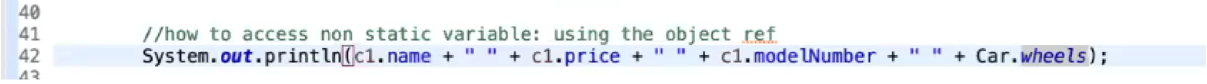
//Cannot make a static reference to the non-static field name



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



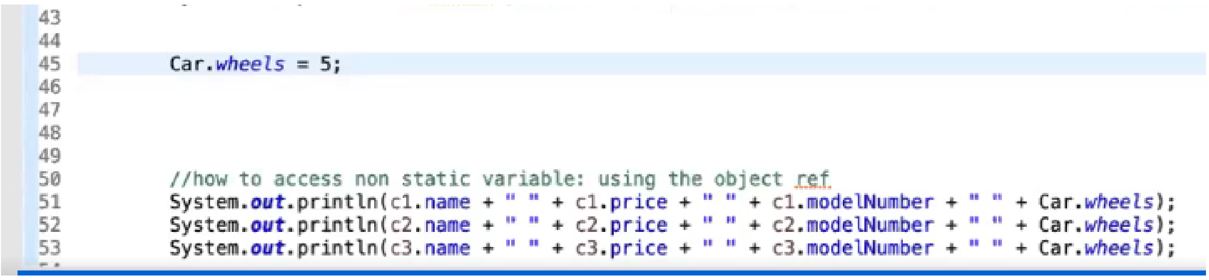
```
        //3. can I access static var using the object ref name?
        System.out.println(c1.wheels);//The static field Car.wheels should be accessed in a static way
```



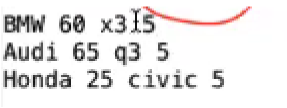
```
        //how to access non static variable: using the object ref
        System.out.println(c1.name + " " + c1.price + " " + c1.modelNumber + " " + Car.wheels);
```

Bmw 6 x3 4

We can change static variables-



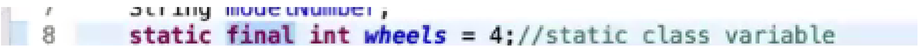
```
        Car.wheels = 5;




        //how to access non static variable: using the object ref
        System.out.println(c1.name + " " + c1.price + " " + c1.modelNumber + " " + Car.wheels);
        System.out.println(c2.name + " " + c2.price + " " + c2.modelNumber + " " + Car.wheels);
        System.out.println(c3.name + " " + c3.price + " " + c3.modelNumber + " " + Car.wheels);
```



```
BMW 60 x3 5
Audi 65 q3 5
Honda 25 civic 5
```

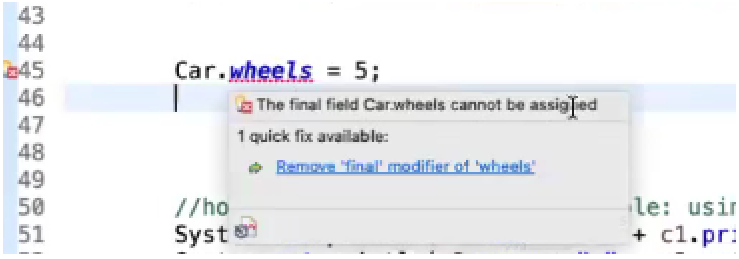
Final-

Makes it constant.



```
    String modelNumber;
    static final int wheels = 4;//static class variable
```

Cant reassign values.



```
        Car.wheels = 5;




        //ho                                    le: usi
        Syst                                    + c1.pr
```

The final field Car.wheels cannot be assigned

1 quick fix available:

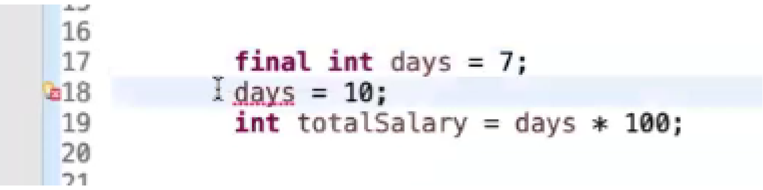
Remove 'final' modifier of 'wheels'

Don’t use static to create constants.

Final can be applied on class and local variables-

when we change local variable which is final.

//The final local variable days cannot be assigned. It must be blank and not using a compound assignment

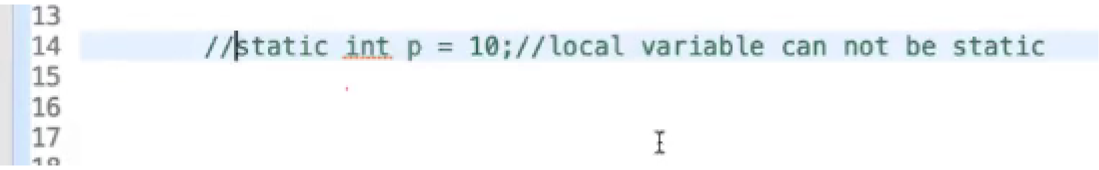


```
        final int days = 7;
        days = 10;
        int totalSalary = days * 100;
```

Static cannot be applied to local variables-

Error we get.

//Illegal modifier for parameter “days”; only final is permitted



```
        //static int p = 10;//local variable can not be static
```

Only class variables can be static.

Note-

We cannot write

Final string name;

We get error.

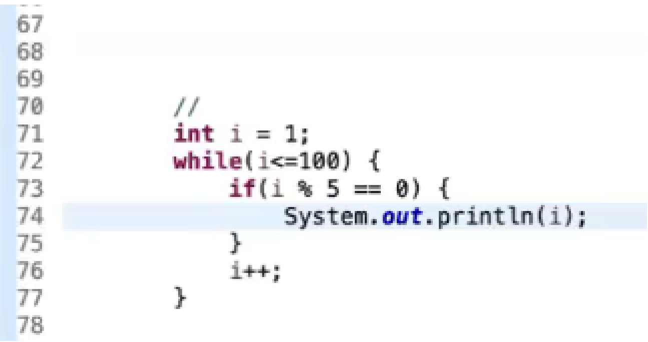
//The blank final field “address” may not have been initialized

We need to assign value like

Final string name =”donnie”;

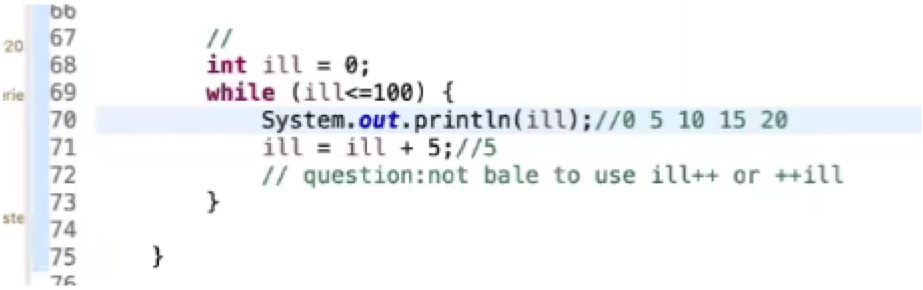
Static –

Only for saving memory.



```
        //
        int i = 1;
        while(i<=100) {
            if(i % 5 == 0) {
                System.out.println(i);
            }
            i++;
        }
```

Prints 5 10 15 20 upto 100.



```
        //
        int ill = 0;
        while (ill<=100) {
            System.out.println(ill);//0 5 10 15 20
            ill = ill + 5;//5
            // question:not bale to use ill++ or ++ill
        }

    }
```

Prints 5 to 100 as earlier.

Note-

Static final int wheels = 4;

In common memory.

Final int days = 7;

Stored in heap memory.

All local variables stored in stack memory.

Note-

Garbage collector only for objects and heap memory.

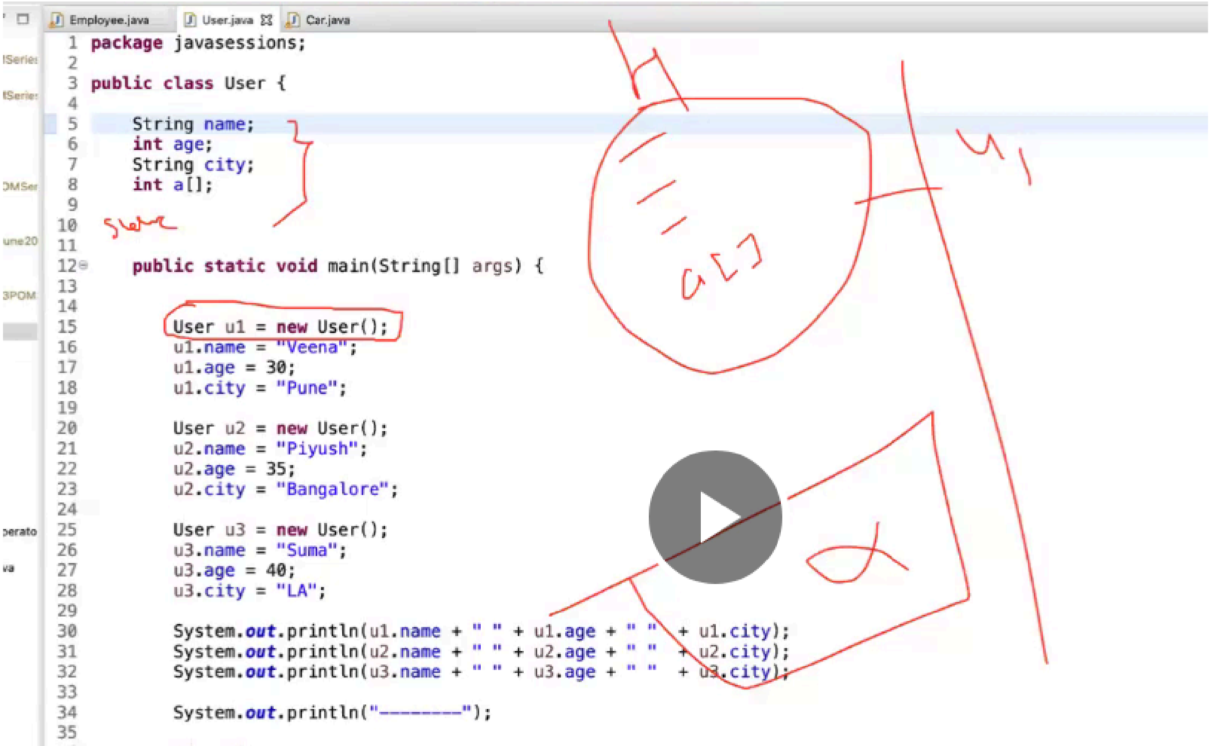
Note-

Static final int wheels =4;

Or final static int wheels = 4;

Both are ok.

Note-



```
package javasessions;

public class User {

    String name;
    int age;
    String city;
    int a[];


    public static void main(String[] args) {

        User u1 = new User();
        u1.name = "Veena";
        u1.age = 30;
        u1.city = "Pune";

        User u2 = new User();
        u2.name = "Piyush";
        u2.age = 35;
        u2.city = "Bangalore";

        User u3 = new User();
        u3.name = "Suma";
        u3.age = 40;
        u3.city = "LA";

        System.out.println(u1.name + " " + u1.age + " " + u1.city);
        System.out.println(u2.name + " " + u2.age + " " + u2.city);
        System.out.println(u3.name + " " + u3.age + " " + u3.city);

        System.out.println("--------");

```

In stack the reference u1 will be created. Every object will have four instance items present.

Common memory is only for ~~stack~~ static items.