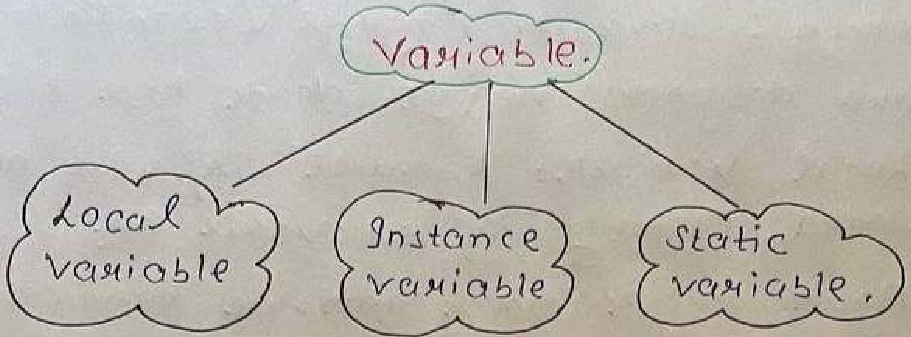
Types of Variable



1. Local variable.

Definition: Jo variables method, constructor, ya kisi block ke under declare hote h.

Characteristics:

- Ye variables sirf usi method ya block ke andar valid hote hain.
- JVM default value nahi deta – initialize karna compulsory hai.

```
ex- {
        String name ; ×
        String Name = "Aditya"; ✓
    }
```

- Access modifier (Public, Private, etc.) nahi lagate hai.
- Memory allocation Stack me hoti hai.

Interview Tip:

Q: What happens if you use a local variable without intializing it?

A: Compile - time - error aayega. JVM local variables ko default value nahi deta.

Q: Why local variables are check at compile-time ?

A: Reason : Java ki language Design Philosophy.

Java designers ne ye decide kiya ki :

- Local variables ka use sirf tab ho jab unka clearly defined value ho.
- Agar programmer intialize karna bhool gaya ho, to compiler hi pakad le - before running the program.
- Agar Java ye galti runtime pe pakdta to bug detect karna mushkil ho jata.

Simple samjho :

" Agar X ko kabhi intialize nahi kiya aur tum uska use kar rahe ho, to ye galti hai, aur isko compiler compile time pe hi, program run karne se phle hi pakad leta hai."

2. Instance variables.

Definition : Jo variable class ke andar declare hote hai, but kisi method ya Constructor ya block ke bahar hote hai.

Characteristics:

- Har Object ka apna alag copy hota hai heap memory me, iska matlab iska value har object me different hota hai. Simple samjho kisi class object hai Obj1 or Obj2, to Obj1 me hum kuch or value assign kar sakte hai or Obj2 me kuch aur.
- JVM default value deta hai agar initialize na kare to. (Java ka ek default value table hota hai - Google kar lena).
- Access modifier laga sakte hai.
- Memory allocation heap me hoti hai, jab object banta hai.
- Object-specific data store karne ke liye use hota hai. (1st point me samjhaya hai).

ex:-
```
Public class Student{
    Private :
        String name;
        int age;
    Public :
        void getData (String n, int a){
            this -> name = n;
            this -> age = a;
        }
}
```

Interview Tip:

Q: When are instance variables created and destroyed?

A: Jab object create hota hai tab instance variable create hote hai, aur jab object destroy hote hai (means jei JVM detect karta hai ki wo object ab use nhi ho raha hai ya phir us object ka value NULL ho jay) to ye bhi destroy ho jata hai automatically via Garbage Collector.

3. Static Variables.

Definition: Jo variable class ke andar declare hote hai aur unke aage "static" keyword lga hota h.

Characteristics:

- Static variable class level pe hote hai, aur pure program me ek hi copy banta hai.
- JVM default value assign kar deta hai.
- Isko object ke through too access kar hi sakte hai, lekin bina object banay class ke name se bhi kar sakte hai. jaise (Classname. var-name).
- Iske pass "this" pointer ka reference nhi hota hai.

example:

```
Class Employee{
    Static String company = "Google";
    String name;
}
```

Interview Tip:

Q: why static variable are not allowed inside a method?

A: Static variable class level variable hote hai, permonent hote hai or ye heap me store hote hai.

Lekin method ke under jo variable hote hai wo local variable hote hai or wo temporary hote hai or wo stack memory me store hote hai.

Isiliye JVM confuse ho jata hai:
"Ek taraf tu keh raha hai static (class-level, peramanent), aur doosri taraf method ke andar bana raha hai (tempo-rary) - ye to logically galat hai"

Conclusion - Static variable sirf class ke block me, aur method ke bahar, declare hote hai. ex -

```
Class Demo{
    Static int count = 0; // Static varable
    void show(){
        int local = 10; // local variable
    }
}
```

Q: What is difference between Static and Instance variables?

A:

Static Variable	Instance Variable.
• Class ke sath associated hota hai.	• Object ke sath associated hota hai.
• Shared accross all Objects	• Har Object ka apna alag copy hota hai.
• Access via Classname. ex- Classname.VarName	• Access via Object.

Q: Can Static variable be overridden?

A: No Static variable belong to the class, not to object. They are not overridden.

Q: can we access instance variables in a static method?

A: No, directly nahi kar sakte. Instance variable ko static method me access karne ke liye object banana prega, aur phir us object ke through access kar sakte hai. ex-

```
Class Demo{
   Public:
        int x = 10; // instance variable
   Static void show (Demo d){ // Static method
            System.Out.Println("value of x:" + d.x);
        }
}
Public Static void main(Strings [] args){
     Demo obj;
     Demo.show (obj); // calling static method using
                         Classname;
}
```