

Object Oriented Programming using Java

Comprehensive overview of Java



sandeepkulange@gmail.com | 9527325202

Session Overview

- Stream associated with Console
- User Input using Scanner
- How to get System Date
- How to solve problem using Object Oriented Paradigm?
- Characteristics of Instance
- Class
- Instance
- Value Type vs Reference Type

Stream Associated with Console

- C Programming Language
 - 1. `stdin`
 - 2. `stdout`
 - 3. `stderr`
- C++ programming language
 - 1. `cin`
 - 2. `cout`
 - 3. `cerr`
 - 4. `clog`
- Java programming language
 - 1. `System.in`
 - 2. `System.out`
 - 3. `System.err`
- `System.in` represents keyboard.
- `System.out` represents Monitor
- `System.err` represents Monitor. Intended for error messages.

User Input using Scanner

- *Scanner* is a final class which is declared in `java.util` package.
- Instantiation:

➤ `Scanner sc = new Scanner(System.in);`

- Methods:

- `public String nextLine();`
- `public int nextInt();`
- `public float nextFloat();`
- `public double nextDouble();`

Example:

```
Scanner sc = new Scanner( System.in );
String name = sc.nextLine();
int empid = sc.nextInt();
float salary = sc.nextFloat();
```

How to get System Date?

- Import `java.util.Date`

```
Date dt = new Date( );
int day = dt.getDate( );
int month = dt.getMonth() + 1;
int year = dt.getYear(); + 1900;
```

- Import `java.util.Calendar`

```
Calendar c = Calendar.getInstance();
int day = c.get(Calendar.DAY_OF_MONTH);
int month = c.get(Calendar.MONTH) + 1;
int year = c.get(Calendar.YEAR);
```

- Import `java.time.LocalDate`

```
LocalDate ldt = LocalDate.now();
int day = ldt.getDayOfMonth();
int month = ldt.getMonthValue();
int year = ldt.getYear();
```

How to solve problem using Object Oriented Paradigm?

- **Problem Statement:** Write a program to accept and print employee record.
- First analyse problem statement and group related data element(s) together.
 1. To group related data elements together define class.

```
class Sample{  
    DataType varName1;          //Non static field  
    static DataType varName2;   //Static field  
}
```

- Non static field is called as instance variable.
- Static field is called as class level variable

```
class Employee{  
    String name;  
    int empid;  
    float salary;  
}
```

How to solve problem using Object Oriented Paradigm?

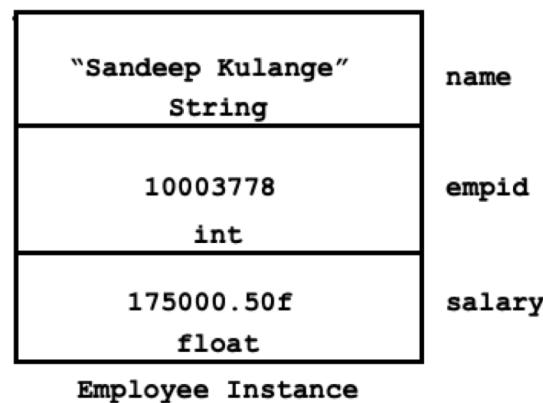
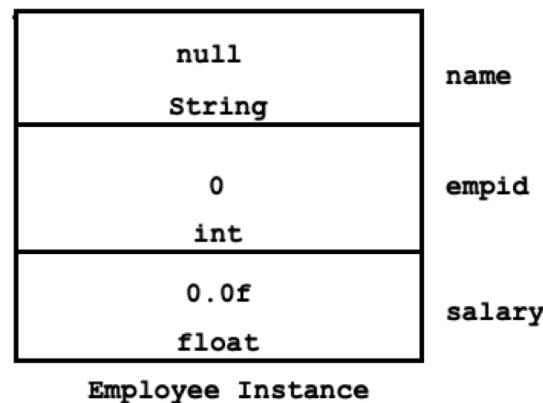
- **Problem Statement:** Write a program to accept and print employee record.
- To store value inside name, empid and salary, it must get space inside memory.
- Since name, empid and salary are non static field declared inside Employee class, it will get space after creating object-instance of the class.
- new is operator in java, which is used to create instance of class on heap section of JVM. Consider below code:

- new Employee(); //Instance of class Employee
- new Employee("Sandeep Kulange", 10003778, 175000.50f); //Instance of class Employe

How to solve problem using Object Oriented Paradigm?

[Java Stack]

[Heap]



How to solve problem using Object Oriented Paradigm?

- **Problem Statement:** Write a program to accept and print employee record.
- If we want to perform some operations on data stored inside instance then we should create reference of it.
- In Java, reference is also called as object reference.
- We can declare reference as a method local variable / field of the class.
- How to declare local reference variable:

```
Employee emp1;
```

```
emp = new Employee( ); //OK
```

```
Employee emp2 = new Employee( "Sandeep Kulange", 10003778, 175000.50f); //OK
```

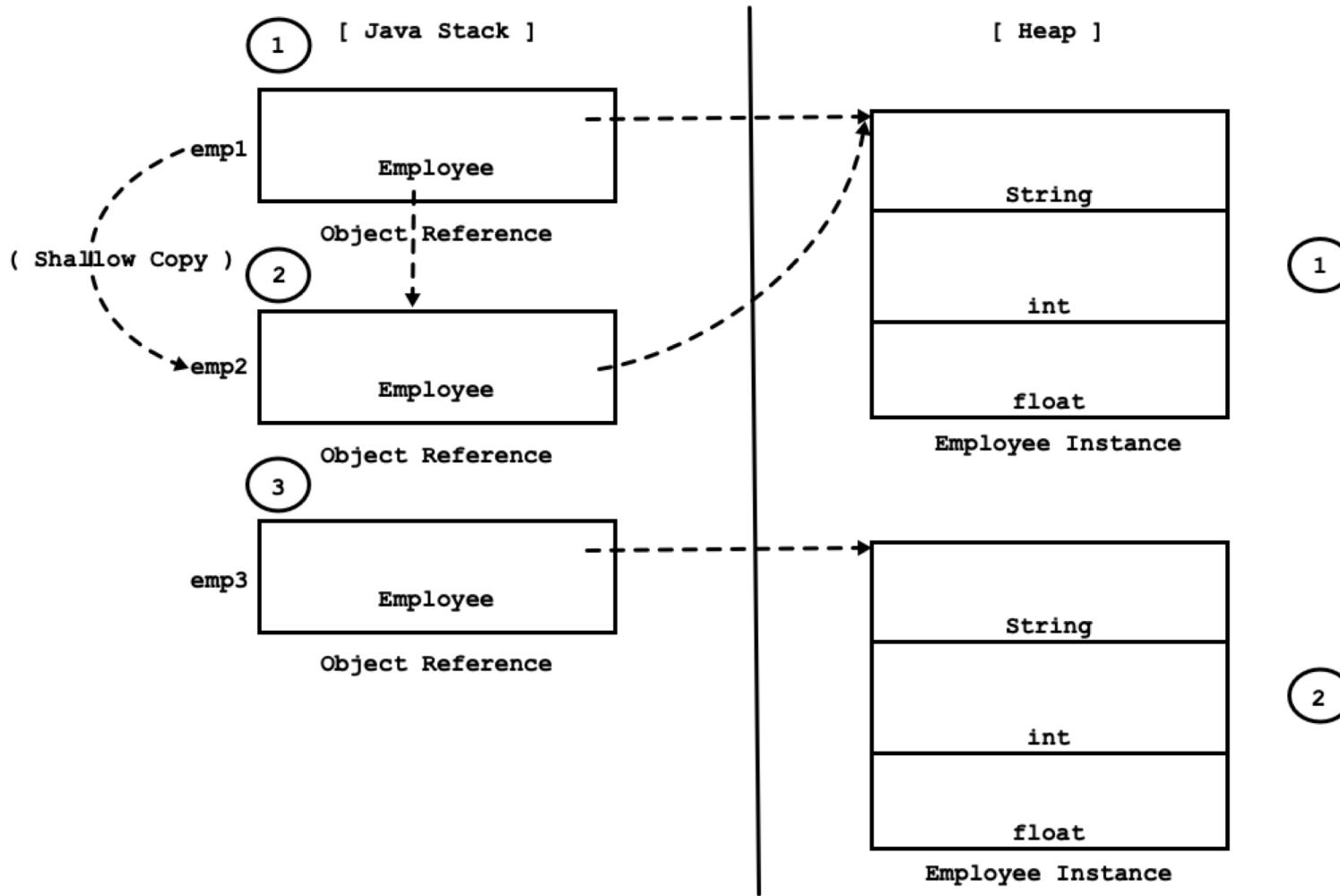
- For more clarity, consider below example. Identify how many instances and how many references?

```
Employee emp1 = new Employee( );
```

```
Employee emp2 = emp1;
```

```
Employee emp3 = new Employee( );
```

How to solve problem using Object Oriented Paradigm?



How to solve problem using Object Oriented Paradigm?

- **Problem Statement:** Write a program to accept and print employee record.
- To process(accept/print) state of the instance, we should call method on it.
- Consider below code:

```
Employee emp = new Employee();
emp.acceptRecord(); //acceptRecord() method is called on emp
emp.printRecord(); //printRecord() method is called on emp
```

- Process of calling method on instance(actually object reference) is called as message passing.

How to solve problem using Object Oriented Paradigm?

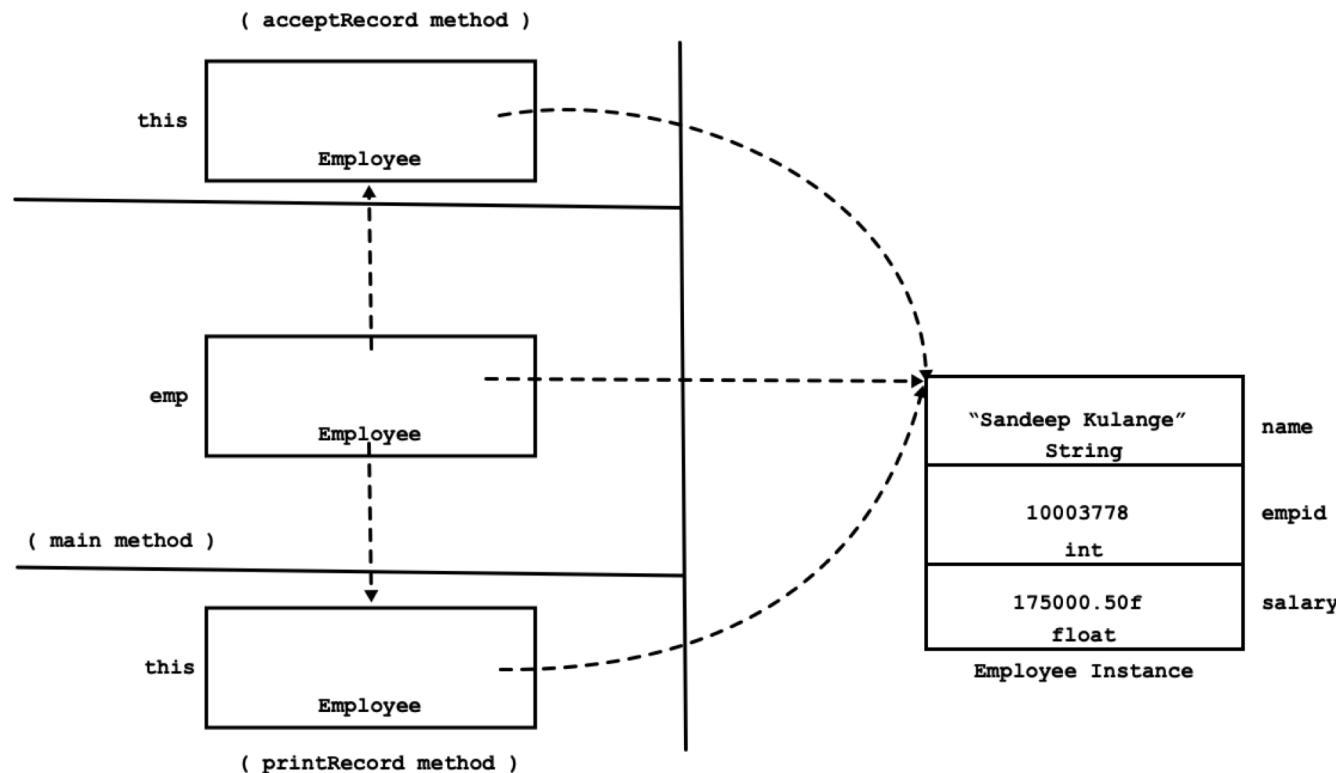
- **Problem Statement:** Write a program to accept and print employee record.
- To call method on instance, first we must define method inside class.
- Function defined inside class is called method. It can be static or non static.
 - Non static methods are designed to call on instance. Hence it is called as instance method.
 - We can call static methods on instance but it is designed to call on type(class/interface) hence it is called as class level method.

```
class Employee{  
    private String name;      //Field  
    private int empid;        //Field  
    private float salary;     //Field  
    void acceptRecord( ){    //Method  
        //TODO  
    }  
    void printRecord( ){     //Method  
        //TODO  
    }  
}
```

```
class Program{  
    public static void main( String[] args ){  
        Employee emp = new Employee( );  
        emp.acceptRecord( );  
        emp.printRecord( );  
    }  
}
```

How to solve problem using Object Oriented Paradigm?

- **Problem Statement:** Write a program to accept and print employee record.
- But now question is how to access non static field/instance variable of the instance inside method?
 - If we call method on instance(actually object reference) then compiler implicitly pass reference of the instance as a argument the method. To catch value of the argument, compiler implicitly declare one parameter is called as this reference.



How to solve problem using Object Oriented Paradigm?

- Instance variable get space once per instance, according to order of non static fields declared inside class.

[Heap]

"Mukesh Salunkhe"	name
String	
10003776	empid
int	
250000.50f	salary
float	

Employee Instance

"Shivprasad Hadpad"	name
String	
10003777	empid
int	
219000.50f	salary
float	

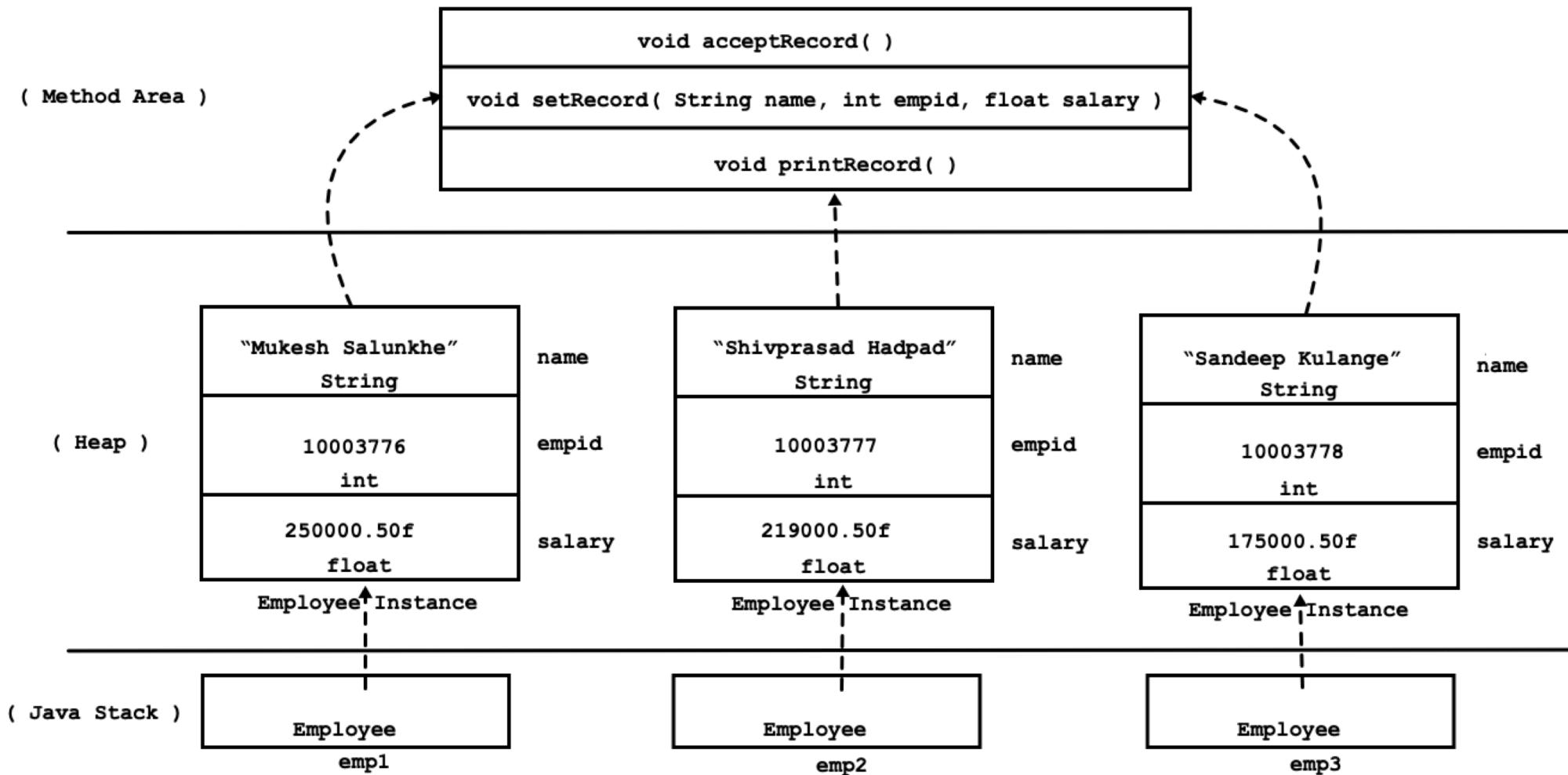
Employee Instance

"Sandeep Kulange"	name
String	
10003778	empid
int	
175000.50f	salary
float	

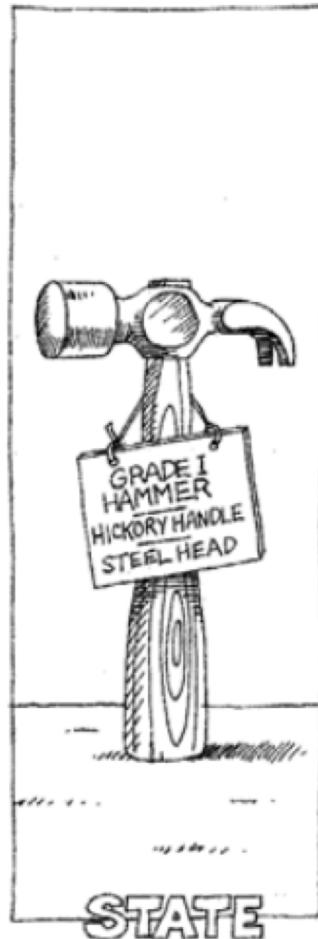
Employee Instance

How to solve problem using Object Oriented Paradigm?

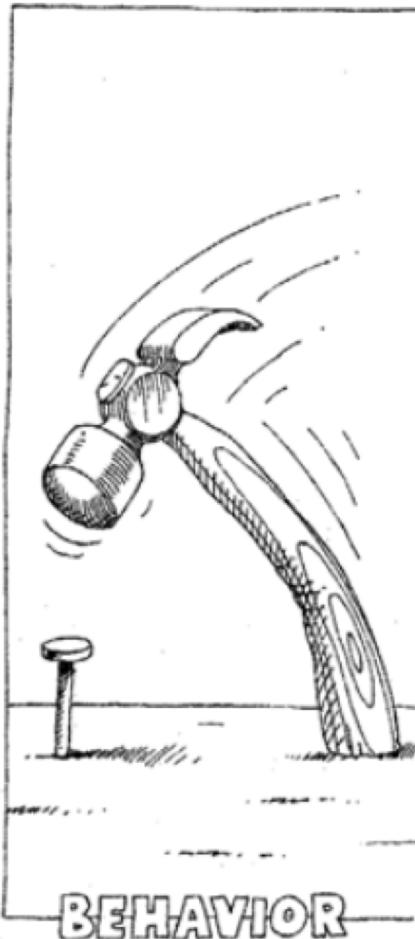
- Method do not get space inside instance. All the instances of same class share its behaviour. This sharing is done by passing reference of the instance to the method.



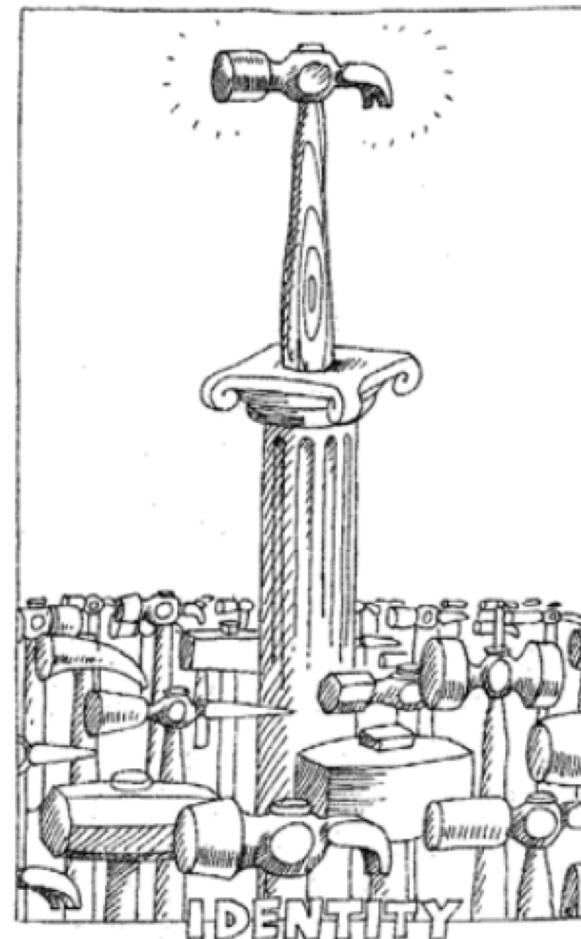
Characteristics of Instance



STATE



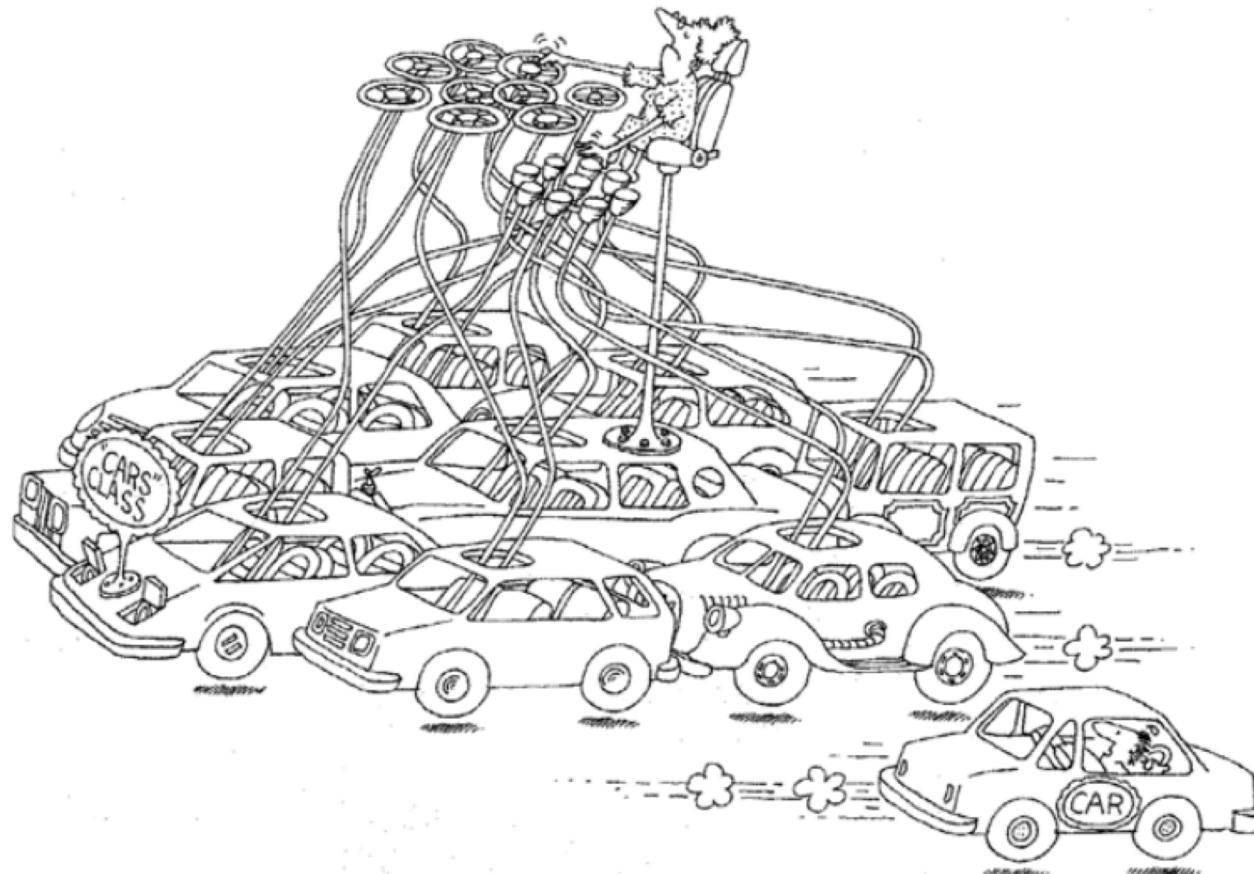
BEHAVIOR



IDENTITY

An object has state, exhibits some well-defined behavior,
and has a unique identity.

Class



Instance



Value Type versus Reference Type

Sr.No.	Value Type	Reference Type
1	Primitive type is also called as value type.	Non primitive type is also called as reference type.
2	boolean, byte, char, short, int, float, double, long are primitive / value types in Java.	Interface, class, enum and array are non primitive / reference types in Java.
3	Variable of value type contains value.	Variable of reference type contains reference.
4	In case of initialization / assignment value gets copied.	In case of initialization / assignment reference gets copied.
5	For the field, default value of primitive /value type variable is zero.	For the field, default value of non primitive /reference type variable is null.
6	Variable of primitive / value type do not contain null value.	Variable of non primitive / reference type can contain null value
7	To create variable of primitive / value type new operator is not required.	To create instance of non primitive type / reference type new operator is required.
8	Variable of value type get space on Java Stack	Instance of reference type get space on Heap.

