# Inheritance (IS-A) vs. Association (HAS-A) Relationship

# Inheritance (IS-A)

- In object-oriented programming, the concept of IS-A is a totally based on Inheritance, which can be of two types Class Inheritance or Interface Inheritance. It is just like saying "A is a B type of thing". For example, Apple is a Fruit, Car is a Vehicle etc. Inheritance is uni-directional. For example, House is a Building. But Building is not a House.
- It is a key point to note that you can easily identify the IS-A relationship. Wherever you see an extends keyword or implements keyword in a class declaration, then this class is said to have IS-A relationship.

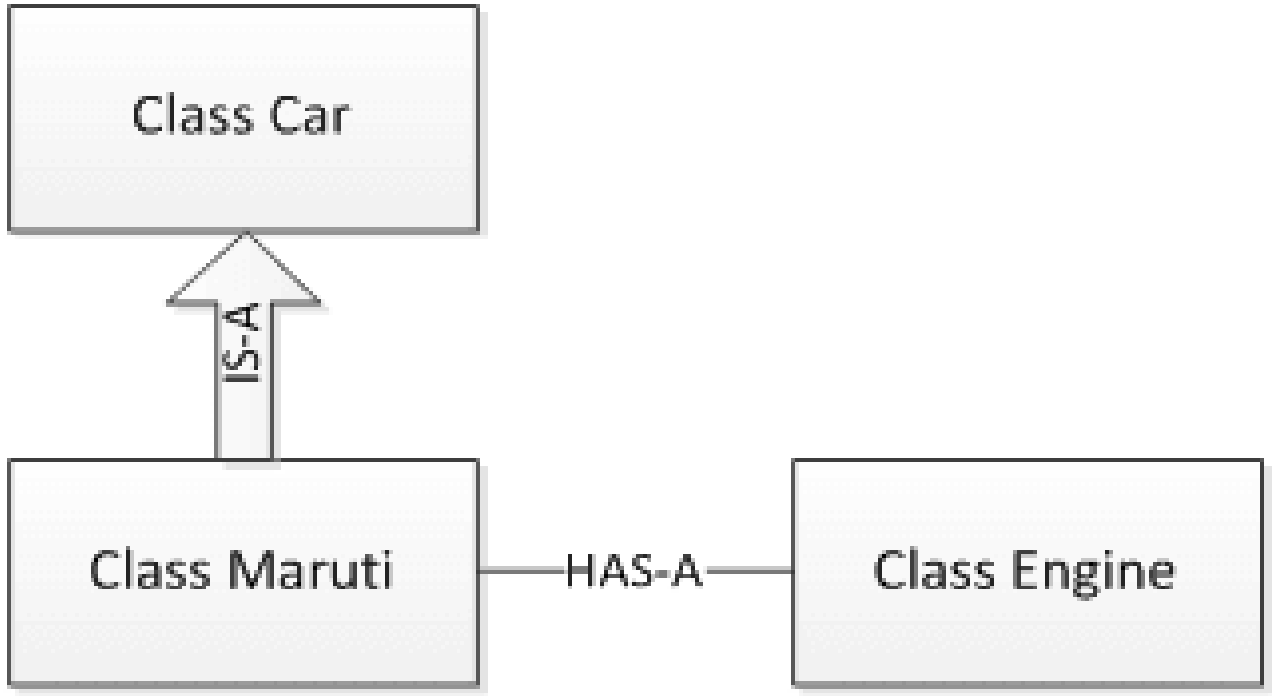
# Association(HAS-A)

- Association in Java is a connection or relation between two separate classes that are set up through their objects. Association relationship indicates how objects know each other and how they are using each other’s functionality. It can be one-to-one, one-to-many, many-to-one and many-to-many

# Association(HAS-A)

- For example, a person can have only one passport. That is a “**one-to-one**” relationship.
- If we talk about the association between a Bank and Employee, a bank can have many employees, So it is a “**one-to-many**” relationship.
- Similarly, every city exists in exactly one state, but a state can have many cities, which is a “**many-to-one**” relationship.
- Lastly, if we talk about the association between a teacher and a student, multiple students can be associated with a single teacher and a single student can also be associated with multiple teachers but both can be created or deleted independently. This is a “**many-to-many**” relationship.

# Inheritance (IS-A) vs. Association (HAS-A) Relationship



```
class Address
{
   int streetNum;
   String city;
   String state;
   String country;
   Address(int street, String c, String
st, String coun)
   {
       this.streetNum=street;
       this.city =c;
       this.state = st;
       this.country = coun;
   }
}
```

```
class StudentClass
{
   int rollNum;
   String studentName;

   Address studentAddr;

       StudentClass(int roll, String name,
Address addr){
       this.rollNum=roll;
       this.studentName=name;
       this.studentAddr = addr;
   }
```

```
public static void main(String args[]){
        Address ad = new Address(55, “Banani", “Dhaka", “Bangladesh");
        StudentClass obj = new StudentClass(123, “Steve Rogers", ad);
        System.out.println(obj.rollNum);
        System.out.println(obj.studentName);
        System.out.println(obj.studentAddr.streetNum);
        System.out.println(obj.studentAddr.city);
        System.out.println(obj.studentAddr.state);
        System.out.println(obj.studentAddr.country);
    }
}
```