

Practice Abstraction

```
import java.io.printstream;
```

```
interface machine {
```

```
PrintStream o = System.out;
```

```
void initialEngine();
```

```
void final Engine();
```

```
}
```

```
abstract class Transport {
```

```
String trademark;
```

```
int manufYear;
```

```
int engine capacity.
```

```
Public Transport (String trademark, int manuf year, int engine  
capacity);
```

```
this. trademark = trademark;
```

```
this. manuf year = manuf year;
```

```
this. engine capacity = engine capacity;
```

```
{
```

```
abstract void print();
```

```
Public void exhibitionInfo () {
```

```
machine. o. println ("Trademark: " + trademark + ",
```

```
Manufacture Year: " + manuf year + ", Engine capacity: " + engine
```

Class extends Transport implement machine of
PublicCar (string trademark, int manufYear, int engineCapacity),
Super (trademark, manufYear, engineCapacity).

@override

public void initial Engine() {

 System.out.println("Car engine starting...");

}

@Override

public void final Engine() {

 System.out.println("Car engine is ending...");

@Override

public void Proper() {

 System.out.println("Car is being driven...");

 System.out.println("Engines = Charged engine. air");

Class Bike extends Transport implement machine of

Public Bike (string trademark, int manufYear,
int engineCapacity);

Super (trademark, manufYear, engineCapacity);

@override

Public void initialEngine()

o. println("Bike engine is starting - - ")

}

override

Public void finalEngine()

o. println(" Bike engine is ending - - ")

}

@override

Public void propell()

o. println(" Bike is being move - - ")

}

class Boat extends Transport implements Machine {

Public Boat (String trademark, int manufYear, int engineCapacity)

Super (trademark, manufYear, engineCapacity)

}

@override

public void initialEngine()

o. println("Boat engine is starting - - ")

@

override

Public void finalEngine()

o. println(" Boat engine is ending - - ")

}

② overriding

public void propel()

o. println ("Boat is being sailed :- ") ;

}

Public class abstraction {

 Public static void main (String [] args) {

 Machine.o. println ("I am Abdule Fahim my ID IT-22031") ;

 Machine . o. println () ;

Transport car = new car ("X-corolla", 2002, 1000),
car. exhibit Info();

((Machine) car). initial Engine ();
car. propel () ;

((Machine) car). final Engine ();

System. Machine . o. println () ;

Transport bike new bike ("Hero Honda", 2001, 150),
bike. exhibit Info();

((Machine) bike). initial Engine ();

bike. propel () ;

((Machine) Bike). final Engine (),

Machine . o. println () ;

TOPIC NAME .

DAY

TIME :

DATE :

Transport boat = new Boat ("Osprey", 2002, 1500);
boat. exhibitInfo();

((Machine) boat). initialEngine();
boat.proper();

((Machine) boat). finalEngine();

}
}

~~Output:~~

Form .

