```
#####################################
package test1;
class Car{

    static {
        System.out.println("----> Class Loading ");
    }

    Car(){
        System.out.println(" ----------> Constructor ");
    }
    void show(){
        System.out.println(" ----------------------> Method of Car class 
");
    }


}
public class Demo1 {
    public static void main(String ss[]){
        try {
            Class.forName("test1.Car");
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}

#####################################
package test2;
class Car{
    static {
        System.out.println(" --> Static Block  ");
    }
    Car(){
        System.out.println(" --------> Constructor ");
    }
    public void show(){
        System.out.println(" ----------------> Method of Car class ");
    }
}
public class Demo1 {
    public static void main(String ss[]){
        try {
            Class c = Class.forName("test2.Car");
            Object ob = c.newInstance();
            Car car = (Car)ob;
            car.show();
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}

#####################################
```

```
package test3;
class Simple{
    void message(){
        System.out.println("Hello Java");
    }
}

class Test{
    public static void main(String args[]){
        try{
            Class c=Class.forName("test3.Simple");
            Simple s=(Simple)c.newInstance();
            s.message();
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
#####################################
package test4;

import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.lang.reflect.Method;

class Sample{
    int x;
    float y;
    char z;

    Sample(){

    }
    Sample(int x){

    }

    void show(){

    }
    void display(){

    }


}
public class Demo1 {

    public static void main(String ss[]) throws Exception{

        Class c=Class.forName("test4.Sample");

        System.out.println("Fields........");
        Field f[]=c.getDeclaredFields();
        for(int i=0;i<f.length;i++)
            System.out.println(f[i]);
        System.out.println("\n\nConstructors........");
        Constructor con[]=c.getDeclaredConstructors();
        for(int i=0;i<con.length;i++)
            System.out.println(con[i]);

        System.out.println("\n\nMethods........");
        Method m[]=c.getDeclaredMethods();
        for(int i=0;i<m.length;i++)
            System.out.println(m[i]);

    }
}

#####################################
class Simple{
}
interface My{
}

class Test{
    public static void main(String args[]){
        try{
            Class c=Class.forName("Simple");
            System.out.println(c.isInterface());

            Class c2=Class.forName("My");
            System.out.println(c2.isInterface());

        }catch(Exception e){
            System.out.println(e);
        }

    }
}
#####################################

public class A {
    private void message(){
        System.out.println("hello java");
    }
}
--------------------------------------------
import java.lang.reflect.Method;
public class MethodCall{
    public static void main(String[] args)throws Exception{

        Class c = Class.forName("A");
        Object o= c.newInstance();
        Method m =c.getDeclaredMethod("message", null);
        m.setAccessible(true);
        m.invoke(o, null);
    }
}

#####################################
```