Assignment 6

Given example for Marker and functional interface.


```java
package Track_Course1;
public class Student implements Cloneable{
    private String name;
    private int age;
    private String department;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getDepartment() {
        return department;
    }
    public void setDepartment(String department) {
        this.department = department;
    }
    public void printDetails() {
        System.out.println("\nEmployee Name: "+name+"\nAge: "+age+"\nDepartment: "+department);

    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /*The class which implements cloneable must
         * override the clone() coming from parent
         * object class.[Cloneable,Serializable,Remote <- Marker Interfaces]
         */
        System.out.println("Cloning");
        Student e = new Student();
        e.setName("Satya");
        e.setAge(22);
        e.setDepartment("Computer science and Engineering");
        e.printDetails();
        try {
            Student cloned = (Student) e.clone(); // it creates a copy of the original object
            cloned.printDetails();
        }
        catch(CloneNotSupportedException e1){
            e1.printStackTrace();
        }
        System.out.println("Finished cloning test");
    }
}
```

Console output:
```
<terminated> Student [Java Application] C:\Program Files\
Cloning

Employee Name: Satya
Age: 22
Department: Computer science and Engineering

Employee Name: Satya
Age: 22
Department: Computer science and Engineering
Finished cloning test
```

```java
package Track_Course1;

interface Driveable{
    default void drive() {
        System.out.println("Drive safely at 120kmph");
    }
}
@FunctionalInterface
interface safely extends Driveable{
    void safely(String msg);
}

public class functionalinterface implements safely {
    public void safely(String msg) {
        System.out.println(msg);
    }


    public static void main(String[] args) {
        functionalinterface f = new functionalinterface();
        f.safely("you should");
        f.drive();
    }
}
```

Console output:
```
<terminated> functionalinterface [Java Ap
you should
Drive safely at 120kmph
```