Problem: 01

```
package Problem1;


public class Animal {
   int id;
   String name;
   int age;

}
public class Mammal extends Animal{
   Mammal(int id,String name,int age){
      this.id=id;
      this.name=name;
      this.age=age;
   }
   void display(){
      System.out.println("Mammal id: "+id);
      System.out.println("Mammal Name:"+name);
      System.out.println("Mammal age: "+age+" Years");
   }
   boolean givesBirth(int n){
      if(n<=2){
         return true;
      }
      else
         return false;
   }

}

public class BirdReptil extends Animal{
   BirdReptil(int id,String name,int age){
      this.id=id;
      this.name=name;
      this.age=age;
   }
   void display(){
      System.out.println("BirdReptil id: "+id);
      System.out.println("BirdRepital Name: "+name);
      System.out.println("BirdRepital age: "+age);
   }
```

```
    boolean laysEggs(int age){
        if(age<=2){
            return true;
        }
        else
            return false;
    }

}
public class Main {
    public static void main(String[] args) {
        Mammal m=new Mammal(221,"Tiya",2);
        m.display();
        if(m.givesBirth(1)==false){
            System.out.println("Not young");
        }
        else{
            System.out.println("Young");
        }

        BirdReptil b=new BirdReptil(12,"Love bird",1);
        b.display();
        if(b.laysEggs(2)==true){
            System.out.println("Egg");
        }
        else{
            System.out.println("Not egg.");
        }


    }

}
```

Output for problem: 01

```
cd /home/gub/NetBeansProjects/LabFinal; JAVA_HOME=
Scanning for projects...

-----------------------< com.mycompany:LabFinal >-
Building LabFinal 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]-----------

--- resources:3.3.0:resources (default-resources)
skip non existing resourceDirectory /home/gub/NetB

--- compiler:3.10.1:compile (default-compile) @ La
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ LabFinal ---
Mammal id: 221
Mammal Name:Tiya
Mammal age: 2 Years
Young
BirdReptil id: 12
BirdRepital Name: Love bird
BirdRepital age: 1
Egg
------------------------------------------------
BUILD SUCCESS
```

Problem: 02

```
package Problem2;

import java.util.logging.Level;
import java.util.logging.Logger;


public class MaxScoreReachedException extends Exception{

   MaxScoreReachedException(String s){
      super(s);
   }

}

public class Player {
```

```
    int n;
    int m;

   synchronized void incrementScore1(){
      n++;
   }
   synchronized void incrementScore(){
      m++;
   }

   void A() throws MaxScoreReachedException{
      if(50<n){
         throw new MaxScoreReachedException("This is MaxScoreReachedException");
      }
      else
      {
         System.out.println(" don't cross limit");
      }
   }

}
public class Main {
    public static void main(String[] args) {
        Player p=new Player();

       Thread t=new Thread(){
          public void run(){
         for(int i=1;i<=40;i++){
          p.incrementScore1();
       }
          }
       };
        t.start();
          try {
            t.join();
        } catch (InterruptedException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        }
          System.out.println("Player 1 Total score: "+p.n);

          try {
           p.A();
        } catch (MaxScoreReachedException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
```

```
        }

         Thread t1=new Thread(){
             public void run(){
          for(int i=1;i<=50;i++){
            p.incrementScore();
        } }
        };

         t1.start();
        try {
            t1.join();
        } catch (InterruptedException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        }

        System.out.println("Player 2 Total score: "+p.n);
        try {
            p.A();
        } catch (MaxScoreReachedException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

}
```

Output for problem: 02

```
--- compiler:3.10.1:compile (default-compile)
Nothing to compile - all classes are up to dat

--- exec:3.1.0:exec (default-cli) @ LabFinal -
Player 1 Total score: 40
 don't cross limit
Player 2 Total score: 40
 don't cross limit
------------------------------------------------
BUILD SUCCESS
------------------------------------------------
Total time:  0.443 s
```

Problem: 03

```
package Problem3;


public interface AudioPlayer {
    void play();
    void pause();
    void stop();
}

public class MP3Player implements AudioPlayer{

    public void play(){
        System.out.println("MP3player music playing...");
    }
    public void pause(){
        System.out.println("MP3player music pause.");
    }
    public void stop(){
        System.out.println("Mp3player music stoping..");
    }

}
public class WAV_Player implements AudioPlayer{
    public void play(){
        System.out.println("Wave player music playing...");
    }
    public void pause(){
        System.out.println("wave player music pasuing..");
    }
    public void stop(){
        System.out.println("wave player music stoping..");
    }

}
public class Main {
    public static void main(String[] args) {
        AudioPlayer a;
        a=new MP3Player();
        a.play();
        a.pause();
        a.stop();
        System.out.println("");
```

```
        a=new WAV_Player();
        a.play();
        a.pause();
        a.stop();
    }

}
```

Output for Problem: 03

```
--- compiler:3.10.1:compile (defaul
Nothing to compile - all classes ar

--- exec:3.1.0:exec (default-cli) @
MP3player music playing...
MP3player music pause.
Mp3player music stoping..

Wave player music playing...
wave player music pasuing..
wave player music stoping..
------------------------------------
BUILD SUCCESS
------------------------------------
```

Problem: 04

```
package Problem4;


public class A {

    static void sum(){
        System.out.println("No parameter.");
    }
    static void  sum(int a,int b){
        System.out.println("sum: "+(a+b));
    }
    static void sum(double a,double b){
        System.out.println("Sum: "+(a+b));
    }
    static void sum(int a,int b,int c){
        System.out.println("Sum: "+(a+b+c));
```

```
    }
}
public class Main {
    public static void main(String[] args) {
        A.sum();
        A.sum(10,20);
        A.sum(10.5,5.5);
        A.sum(5,10,20);
    }

}
```

Output for Problem: 04