1.Bike

```
package homework1_interface_abstract;

public class Bike extends Thing {
	private int age;
	private boolean isFrontBreakWorking;
	private boolean isBackBreakWorking;

	public Bike(String name, int age, boolean isFrontBreakWorking, boolean
isBackBreakWorking) {
		super(name);
		this.age = age;
		this.isBackBreakWorking  =  isBackBreakWorking;
		this.isFrontBreakWorking = isFrontBreakWorking;
	}

	public boolean isValid() {
		boolean isValids = false;
		if (age < 2 && isFrontBreakWorking || isBackBreakWorking) {
			return true;
		} else {
			return isValids;
		}

	}
	@Override
	public String toString() {
		String message = "";
		if(isValid()) {
			message = "isValid";
		}else {
			message = "InValid";
		}
		return super.toString() +  message;
	}
}
```

2. ItTeacher

```
package homework1_interface_abstract;

public class ItTeacher extends Thing {
	private int yearsOfExperience;
	
	public ItTeacher(String name,int yearsOfExperience ) {
		super(name);
		this.yearsOfExperience = yearsOfExperience;
	}
	public boolean isValid() {
		boolean isBackBreakWorking = false;
		if (yearsOfExperience > 2) {
			return true;
		}else {
			return false;
		}
	}
	
	@Override
	public String toString() {
		String message = "";
		
		if(isValid()) {
			message = "isValid";
		}else {
			message = "InValid";
		}
		return super.toString() +  message;
	}

}
```

3. Main

```
 package homework1_interface_abstract;


import java.util.ArrayList;
import java.util.List;


public class Main {
    public static void main(String[] args) {
        List<Thing> allThings = new ArrayList<>();
        // Object of class Bike
        allThings.add(new Bike("bike1", 1, false, true)) ;
        allThings.add(new Bike("bikes2", 7, true, false));
        // Object of class ItTeacher
        allThings.add(new ItTeacher("Rady", 5));
        allThings.add(new ItTeacher("Rith", 1));
        // Object of class RiceCooker
        allThings.add(new RiceCooker("khamyoeung",5, 500));
        allThings.add(new RiceCooker("sokha",3, 400));
          // Loop for arrays list
        for (int i = 0; i < allThings.size(); i++) {
            System.out.println(allThings.get(i));
        }


    }


}
```

4. RiceCooker

```
package homework1_interface_abstract;

public class RiceCooker extends Thing {
       private int age;
       private int numberOfUse;
       public RiceCooker(String name, int age, int numberOfUse) {
             super(name);
             this.age = age;
             this.numberOfUse = numberOfUse;
       }
       public boolean isValid() {
             if(age < 5 && numberOfUse < 500) {
                    return true;
             }else {
                    return false;
             }
       }

       @Override
       public String toString() {
             String message = "";
             if(isValid()) {
                    message = "isValid";
             }else {
                    message = "InValid";
             }
             return super.toString() +  message;
       }
}
```

5. Thing

```
package homework1_interface_abstract;

public abstract class Thing implements Validable {
    String name;
    public Thing(String name) {
       this.name = name;
       }
    public String toString() {
             return name + " ";
       }

}
```

6. Validable interface

```
package homework1_interface_abstract;

public interface Validable {
    boolean isValid();
}
```

II.homework2

1. EquilateralTriangle

```
package homework2_interface_abstract;

public class EquilateralTriangle extends Polygon implements RegularPolygon{
    int getSideLength;
    final int getNumSides = 3;
    public EquilateralTriangle(int getSideLength) {
         super(getSideLength);
    }

    @Override
    public int getNumSides() {
        return getNumSides;
    }

    @Override
    public int getSideLength() {
        return getSideLength;
    }

}
```

2. Main

```
package homework2_interface_abstract;

import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        ArrayList<RegularPolygon> array = new ArrayList<RegularPolygon>();
      // create Object of square
        Square squar = new Square(0);
        EquilateralTriangle Triangle = new EquilateralTriangle(6);
        array.add(squar);
        array.add(Triangle);
        System.out.println("Numsides Of regularPolygon is:" + 
RegularPolygon.getTotalSides(array));

    }

}
```

3. Polygon

```
package homework2_interface_abstract;
public abstract class Polygon{
    public int getSideLength;
    public Polygon(int getSideLength) {
        this.getSideLength = getSideLength;
    }
}
```

4. egularPolygon

```
package homework2_interface_abstract;


import java.util.ArrayList;
import java.util.List;


public interface RegularPolygon {


        int getNumSides();
        int getSideLength();
        public static int getTotalSides(ArrayList<RegularPolygon> array) {
            int sumOfRegularPolygon = 0;
            for (RegularPolygon regularPolygonOfArrays : array) {
                sumOfRegularPolygon = sumOfRegularPolygon + regularPolygonOfArrays.getNumSides();
            }
            return sumOfRegularPolygon;
        }
}
```

5. Square

```
package homework2_interface_abstract;

public class Square extends Polygon implements RegularPolygon {

        final int getNumSides = 4;
        public Square(int getSideLength) {
                super(getSideLength);
        }

        @Override
        public int getNumSides() {
                return getNumSides;
        }

        @Override
        public int getSideLength() {
                return getSideLength;
        }
}
```