COMP 110/401
LEAST PRIVILEGE

Instructor: Prasun Dewan
PREREQUISITE

- Interfaces
public class ABMISpreadsheet implements BMISpreadsheet {
    double height, weight, bmi;
    ...
}
public class ABMISpreadsheet implements BMISpreadsheet {
    public double height, weight, bmi;
    ...
}
Hard to Change

```java
public class ABMISpreadsheet implements BMISpreadsheet {
    public double height, weight;
    ...
}
```

Other classes can access
**Consistency Constraints Violated**

```java
public class ABMISpreadsheetWithPublicVariables {
    public double height, weight, bmi;
    ...
}
```

```java
bmiSpreadsheet = new ABMISpreadsheetWithPublicVariables();
bmiSpreadsheet.weight = 75;
bmiSpreadsheet.height = 1.77;
bmiSpreadsheet.bmi = 1.2;
```
public class ABMISpreadsheet implements BMISpreadsheet {
    public double height, weight, bmi;
    ...
}

More on this later
ENCAPSULATION PRINCIPLE

- Do not make instance variables public
  - Expose them through public methods
**CONSTANTS TYPICALLY SHOULD BE PUBLIC**

Inconsistent value cannot be stored

```java
public interface BMISpreadsheet {
    public final double CMS_IN_INCH = 2.54;
    public final double LBS_IN_KG = 2.2;
    ...
}
```

Implementation independent

Accessible to all implementing classes
PRINCIPLE

- Declare implementation-independent named constants in interfaces
  - implementing classes can access them
```java
public class AnotherBMISpreadsheet implements BMISpreadsheet {
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = weight/(height*height);
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = weight/(height*height);
    }
    public double getBMI() {
        return bmi;
    }
}
```

Assuming ABMIColorator does not exist
public class AnotherBMISpreadsheet implements BMISpreadsheet{
  double height, weight, bmi;
  public double getHeight() {
    return height;
  }
  public void setHeight(double newHeight) {
    height = newHeight;
    bmi = calculateBMI();
  }
  public double getWeight() {
    return weight;
  }
  public void setWeight(double newWeight) {
    weight = newWeight;
    bmi = calculateBMI();
  }
  double calculateBMI() {
    return weight/(height*height);
  }
  ....
}
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = calculateBMI();
    }

    public double getWeight() {
        return weight;
    }

    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = calculateBMI();
    }

    double calculateBMI() {
        return (weight/2.2)/(height * 2.54/100*height*2.54/100);
    }
    
    ...
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;

    ...

    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = calculateBMI();
    }

    public double getWeight() {
        return weight;
    }

    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = calculateBMI();
    }

    double calculateBMI() {
        return (weight/2.2)/(height * 2.54/100*height*2.54/100);
    }

    ...
}
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = calculateBMI();
    }
    ...
    double calculateBMI() {
        return (weight/2.2)/(height * 2.54/100);
    }
    ...
}
Do not give a user of some code more rights than it needs
- Code is easier to change
- Need to learn less to use code
- Less likelihood of accidental or malicious damage to program

Like hiding engine details from car driver
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    ...
    final double LBS_IN_KG = 2.2;
    final double CMS_IN_INCH = 2.54;
    ...
    double calculateBMI() {
        return (weight/LBS_IN_KG) /
               (height*CMS_IN_INCH/100*height*CMS_IN_INCH/100);
    }
    ...
}
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    ...
    final double LBS_IN_KG = 2.2;
    final double CMS_IN_INCH = 2.54;
    ...
    double calculateBMI() {
        double heightInMeters = height*CMS_IN_INCH/100;
        return (weight/LBS_IN_KG) /
                (heightInMeters*heightInMeters);
    }
    ...
}
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    double heightInMeters;

    ...
    final double LBS_IN_KG = 2.2;
    final double CMS_IN_INCH = 2.54;
    ...
    double calculateBMI() {
        heightInMeters = height*CMS_IN_INCH/100;
        return (weight/LBS_IN_KG) /
            (heightInMeters*heightInMeters);
    }
    ...
}
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    double heightInMeters = height*CMS_IN_INCH/100;
    ...
    final double LBS_IN_KG = 2.2;
    final double CMS_IN_INCH = 2.54;
    ...
    public void setHeight(double newHeight) {
        heightInMeters = newHeight;
        bmi = calculateBMI();
    }
    ...
    double calculateBMI() {
        return (weight/LBS_IN_KG) /
            (heightInMeters*heightInMeters);
    }
    ...
}
public class AnotherBMISpreadsheet implements BMISpreadsheet{

double height, weight, bmi;

...

public void setHeight(double newHeight) {
    height = newHeight;
    bmi = calculateBMI();
}

public double getWeight() {
    return weight;
}

public void setWeight(double newWeight) {
    weight = newWeight;
    bmi = weight/(height*height);
}

...

double calculateBMI () {
    double heightInMetres = height*CMS_IN_INCH/100;
    return (weight/LBS_IN_KG) / (heightInMetres*heightInMetres);
}

...}
**Scope of Public Items**

```java
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;
    ...
    public double getWeight() {
        return weight;
    }
    ...
}
```

- ObjectEditor
- ABMISpreadsheet
**Scope Modifiers**

- **public**: accessible in all classes.
- **protected**: accessible in all subclasses of its class and all classes in its package.
  - Will see this later.
  - Many of the variables/methods in lecture code have protected access even though PPT slides do not show it.
- **default**: accessible in all classes in its package.
- **private**: accessible only in its class.

Will use default access for non public variables as we do not know the full context for the code right now.

Some purists of least privilege insist on private access.
IDENTIFIER SCOPE

- Region of code where the identifier is visible
- Arbitrary scopes not possible
- Least Privilege => Make scope as small as possible
public class AnotherBMISpreadsheet implements BMISpreadsheet{
    double height, weight, bmi;

    ... public void setHeight(double newHeight) {
        height = newHeight;
        bmi = calculateBMI();
    }

    public double getWeight() {
        return weight;
    }

    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = weight/(height*height);
    }

    ... double calculateBMI() () {
        double heightInMetres = height*CMS_IN_INCH/100;
        return (weight/LBS_IN_KG) / (heightInMetres*heightInMetres);
    }

    ...}

FOLLOWING LEAST PRIVILEGE

...
public class ABMISpreadsheet {
    double height;
    double weight;
    public ABMISpreadsheet(
        double theInitialHeight, double theInitialWeight) {
        setHeight(theInitialHeight);
        setWeight(theInitialWeight);
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
    }
    ...
}
public class ABMISpreadsheet {
    double height;
    double weight;
    public ABMISpreadsheet(
        double height, double weight) {
        setHeight(height);
        setWeight(weight);
    }
    public void setWeight(double weight) {
        weight = weight;
    }
    public void setHeight(double height) {
        height = height;
    }
    ...
}
Disambiguation with this (Standard Convention)

```java
class ABMISpreadsheet {
    double height;
    double weight;
    public ABMISpreadsheet(
        double height, double weight) {
        setHeight(height);
        setWeight(weight);
    }
    public void setWeight(double weight) {
        this.weight = weight;
    }
    public void setHeight(double height) {
        this.height = height;
    }
    ...
}
```

- Local, not global instance variable
- Eclipse features based on this convention
- Can forget to put this
public class ABMISpreadsheet {
    double height;
    double weight;
    public ABMISpreadsheet(
        double theHeight, double theInitialWeight) {
        setHeight(theHeight);
        setWeight(theInitialWeight);
    }
    public void setWeight(double aWeight) {
        weight = weight;
    }
    public void setHeight(double newVal) {
        height = newVal;
    }
    ...
}

Using Different Names

Must sometimes fight with Eclipse

Examples use multiple conventions for local variables