# CISH-6510 Web Application Design and Development

# Overview of JavaBeans

## Overview

- What Are JavaBeans?
- `WeatherBean` Example
- Bean Categories

2

## What Are JavaBeans?

- Components written in Java.
- JavaBeans API provides standard format for Java classes.
- JavaBean: class that follows the JavaBeans API.
  - May be packaged with support files in a JAR file
- Bean container: application, environment, or programming language that allows beans to be manipulated.

3

## What Are JavaBeans? (cont.)

- Three ways for beans to interact:

1. Properties: Conceptually equivalent to attributes.

- All properties must be private.
- May be single value or an array.
- May be Java or user-defined type.
- Properties represent bean’s behavior and appearance.
  - E.g., clock bean would have time zone and alarm properties

4

## What Are JavaBeans? (cont.)

- **Accessible properties must be have `get` and `set` methods.**
  - **Must be written by developer**
- **Bean's capabilities are documented in a table called a property sheet.**
  - **Lists all properties available on the bean**
  - **Allows bean designers to describe features of a bean to its users**

**5**

## What Are JavaBeans? (cont.)

| **Name** | **Access** | **Type** | **Example** |
|---|---|---|---|
| Zipcode | R/W | String | 06120 |
| current Temp | R | int | 70 |
| high | R | int | 80 |
| low | R | int | 50 |
| forecast | R | String | Sunny |

**6**

## What Are JavaBeans? (cont.)

2. **Methods:**

- Any public Java method is an available bean method.
- Methods less important in Beans than in normal Java classes.
  - Most functionality located in properties and events
- Methods are only a secondary way for clients to interact.
- Developer must provide `get` and `set` methods.

7

## What Are JavaBeans? (cont.)

```
public String getZipcode()
        {  return zipcode; }

public void setZipcode(
      String zip)
        {  zipcode = zip; }
```

8

## What Are JavaBeans? (cont.)

- `Booleans` have a special accessor method:

```
public boolean isAttribute()
 { return attribute; }
```

- Developer must provide a zero-argument constructor.
  - Java provides one, but if you define a constructor with parameters, Java’s zero-argument constructor disappears

9

## What Are JavaBeans? (cont.)

3. <u>Events:</u>

- Beans can communicate with each other via firing and receiving events.
- Beans can be defined as “listeners”.
- Other beans send notifications to listeners that register an interest in being noticed when the event fires.
- Event source beans provide pair of methods to add or remove listeners from event notification list.

10

## What Are JavaBeans? (cont.)

```
public void
   addEventListenerType(
      EventListenerType e);

public void
   removeEventListenerType(
       EventListenerType e);
```

11

## WeatherBean Example

```
package heidic;
public class WeatherBean
{// Define private properties
  private String zipcode;
  private int    currentTemp;
  private int    high;
  private int    low;
  private String forecast;
```

12

## WeatherBean Example (cont.)

```
public WeatherBean()
{ zipcode = "00000";
  currentTemp = 0;
  high = 0;
  low = 0;
  forecast = "Not available.";
}
```

13

## WeatherBean Example (cont.)

```
public String getZipcode()
{  return zipcode;    }

public void setZipcode(
   String zip)
{  zipcode = zip;     }

public int getCurrentTemp()
{  return currentTemp;    }
```

14

## WeatherBean Example (cont.)

```
public int getHigh()
{  return high;      }

public int getLow()
{  return low;       }

public String getForecast()
{  return forecast;  }
```

15

## WeatherBean Example (cont.)

```
public void update(String z)
{ zipcode = z;
  if (zipcode.equals("06120"))
  {
   currentTemp = 70;
   high = 72;
   low = 50;
   forecast = "Cloudy";
 }
```

16

## WeatherBean Example (cont.)

```
else if(zipcode.equals("11111"))
 { currentTemp = 30;
   high = 32;
   low = 10;
   forecast = "Snowy";
}else if(zipcode.equals("22222"))
 {  currentTemp = 90;
    high = 92;
    low = 80;
    forecast = "Sunny"; }
```

17

## WeatherBean Example (cont.)

```
else {
   currentTemp = 70;
   high = 72;
   low = 50;
   forecast = "Cloudy"; }
 }
}
```

18

## Bean Categories

- Three general categories of beans:

1. <u>Visual Component Beans:</u>

- Elements such as textfields, selectors, useful for building UIs.
- One of most common uses of JavaBeans.
- Visual beans are not compatible with servlets/JSP.
  - Visual beans are graphic, servlets/JSPs are textual-based

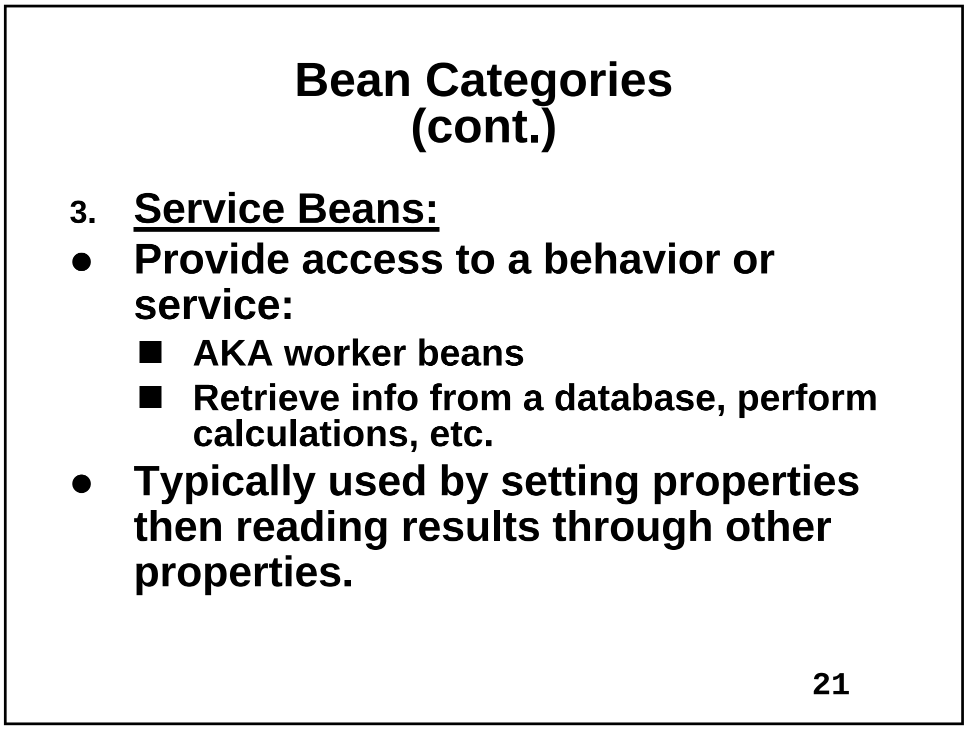
19

## Bean Categories (cont.)

2. <u>Data Beans:</u>

- Convenient way to hold data.
- Calculation responsibility of some other component.
- Data beans are typically read-only.
  - Loaded upon creation
- Useful to standardize access to information by providing stable interface.

20



## Bean Categories (cont.)

3. **<u>Service Beans:</u>**

- **Provide access to a behavior or service:**
  - ■ **AKA worker beans**
  - ■ **Retrieve info from a database, perform calculations, etc.**
- **Typically used by setting properties then reading results through other properties.**

21