



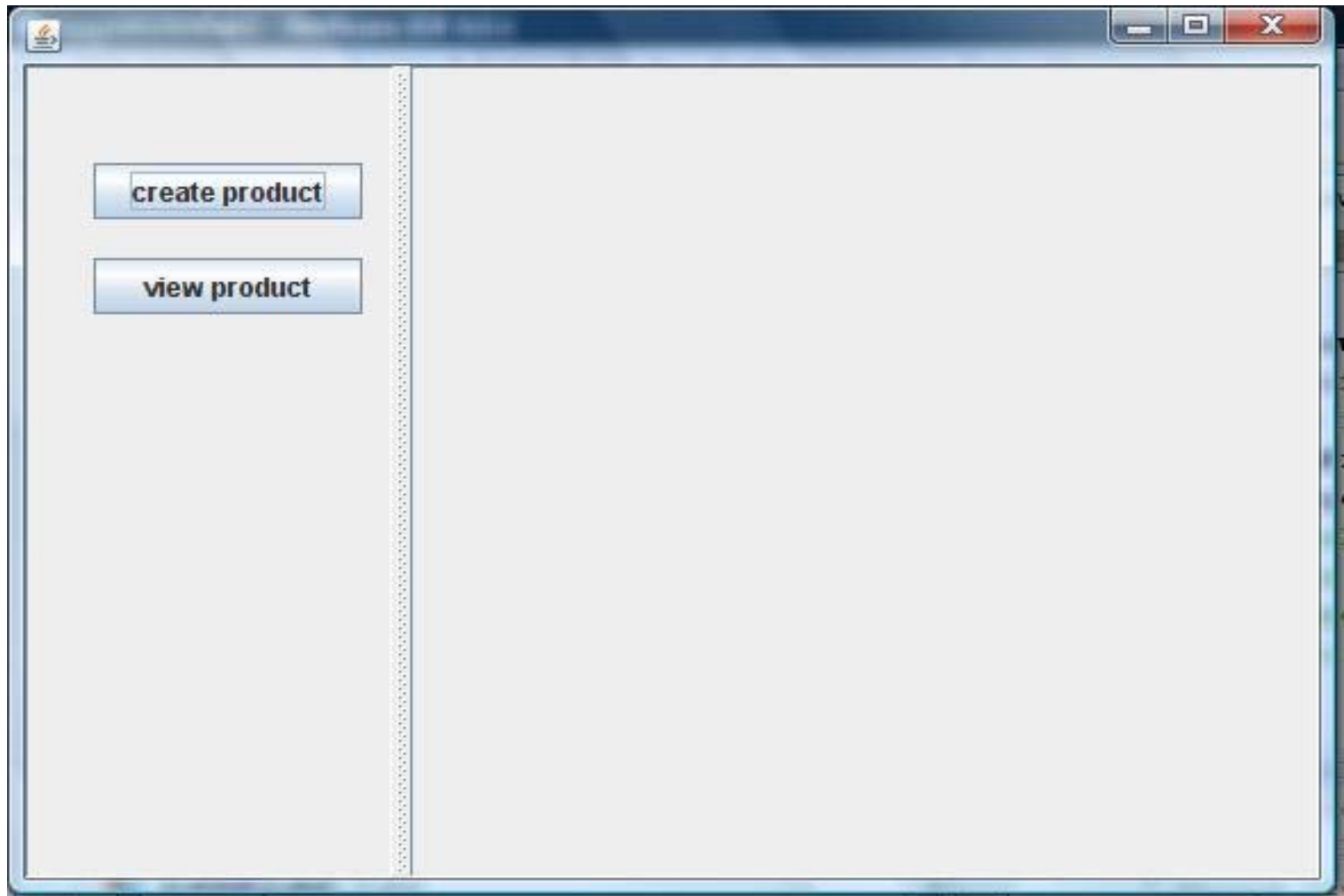
ISYG 100 :Application Engineering and Development

Lab 1 : Class and Object

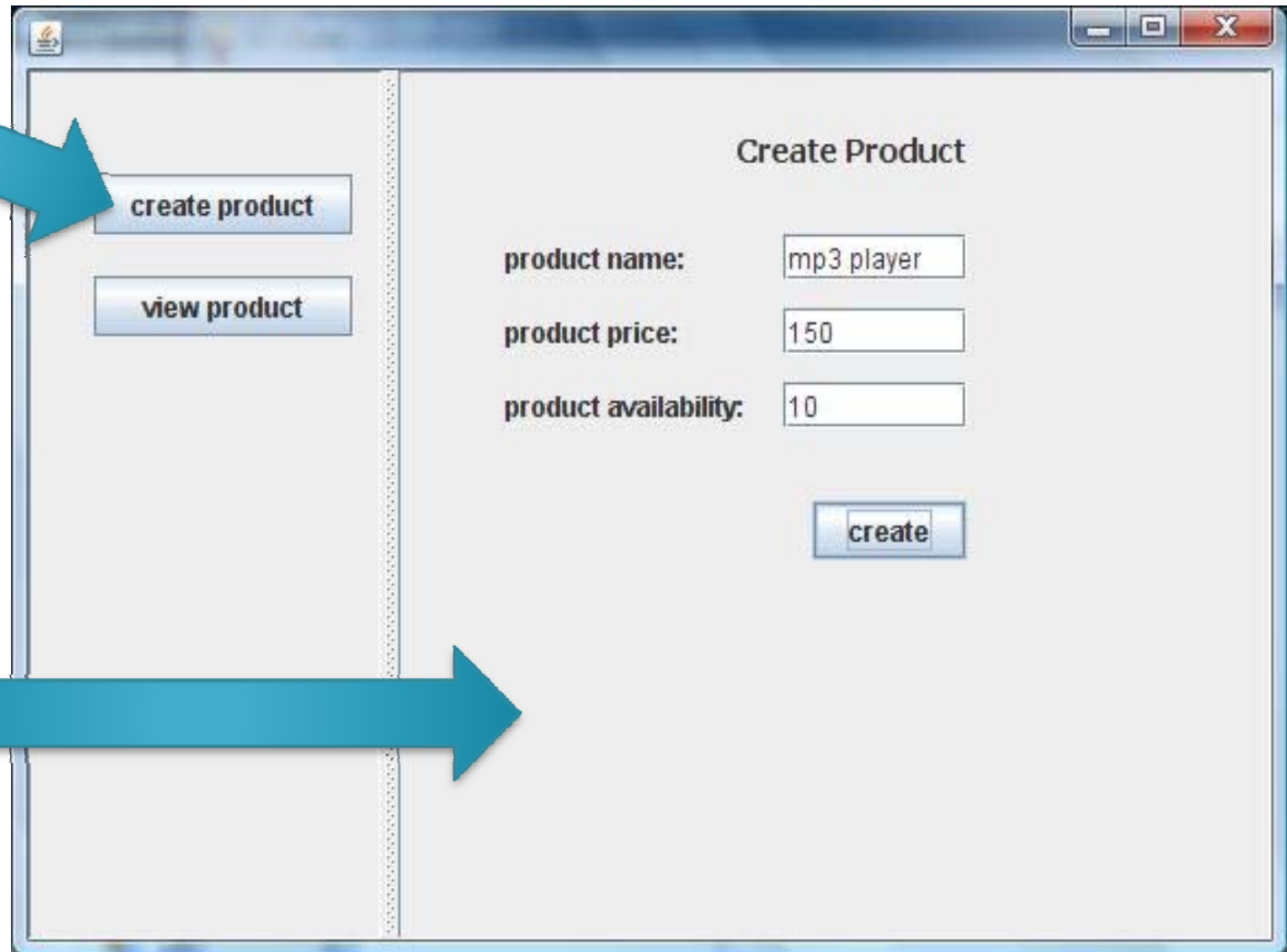
Objective

- Demonstrate how to create a java/swing application
- How to define java classes
- How to create and populate java objects
- How to pass data between from the JFrame to JPanels

Output Application I



Output Step I



Create Product

create product

view product

product name:

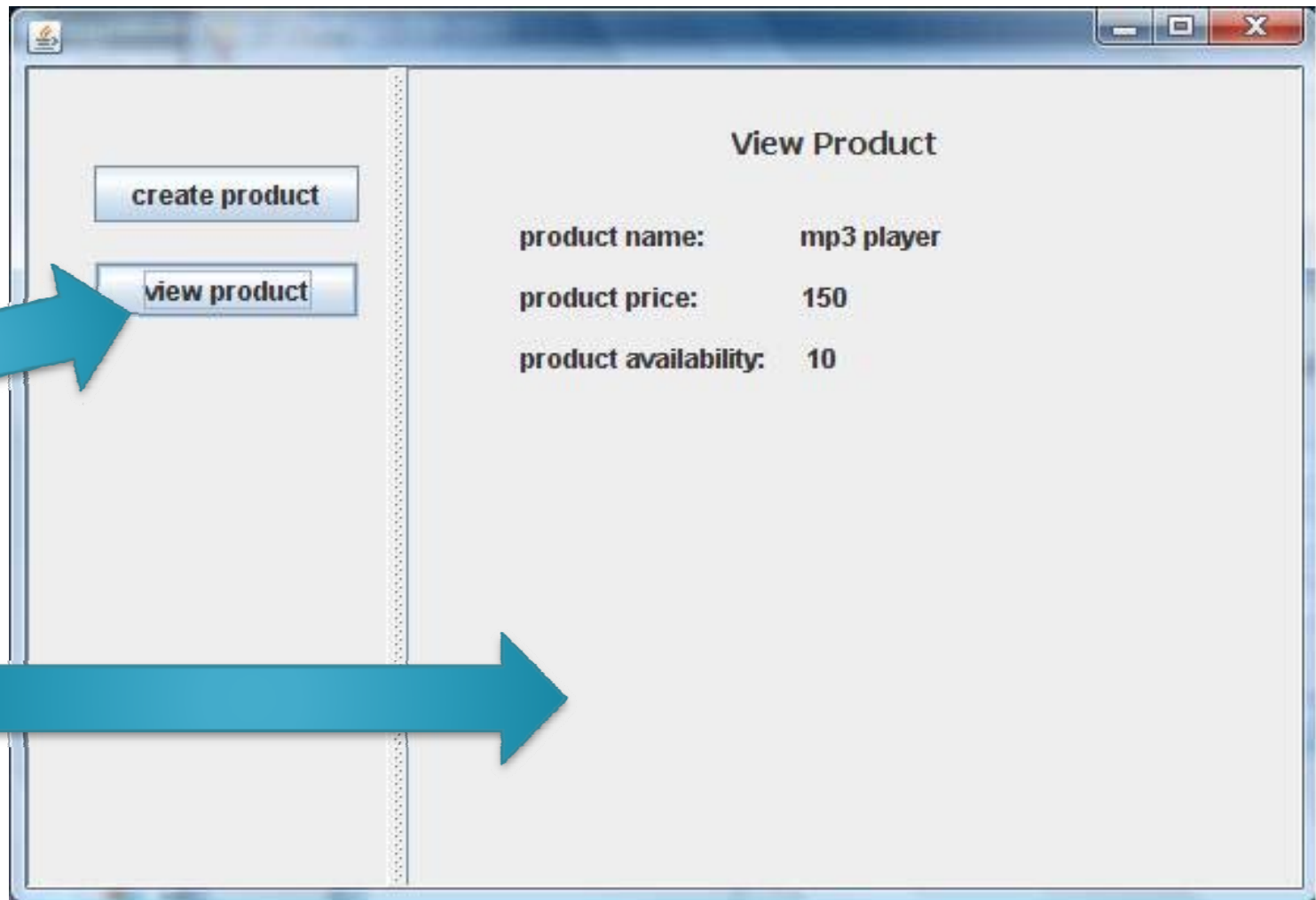
product price:

product availability:

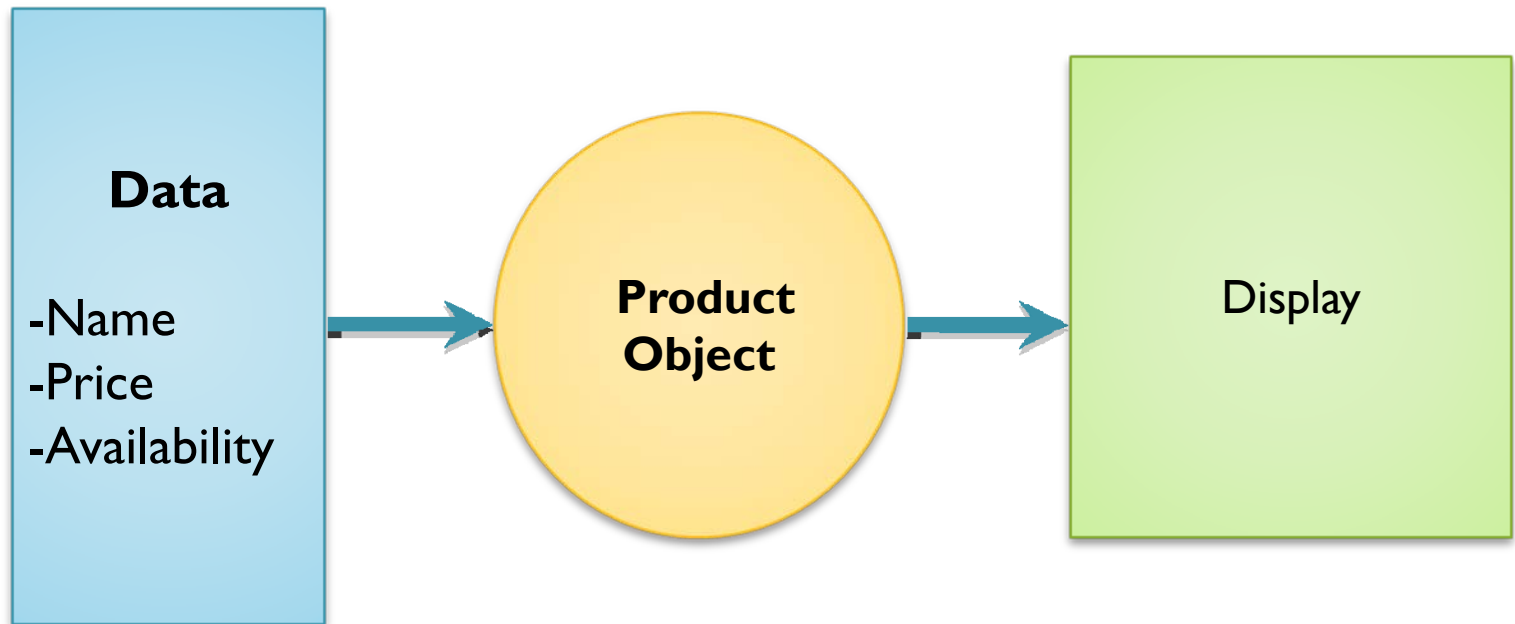
create

The image shows a software window titled 'Create Product'. On the left side, there are two buttons: 'create product' and 'view product'. On the right side, there are three input fields labeled 'product name:', 'product price:', and 'product availability:'. The 'product name' field contains 'mp3 player', the 'product price' field contains '150', and the 'product availability' field contains '10'. Below these fields is a 'create' button. Two blue arrows are overlaid on the image: the first arrow, labeled with a large '1' in a blue circle, points to the 'create product' button; the second arrow, labeled with a large '2' in a blue circle, points from the left side of the window towards the right side.

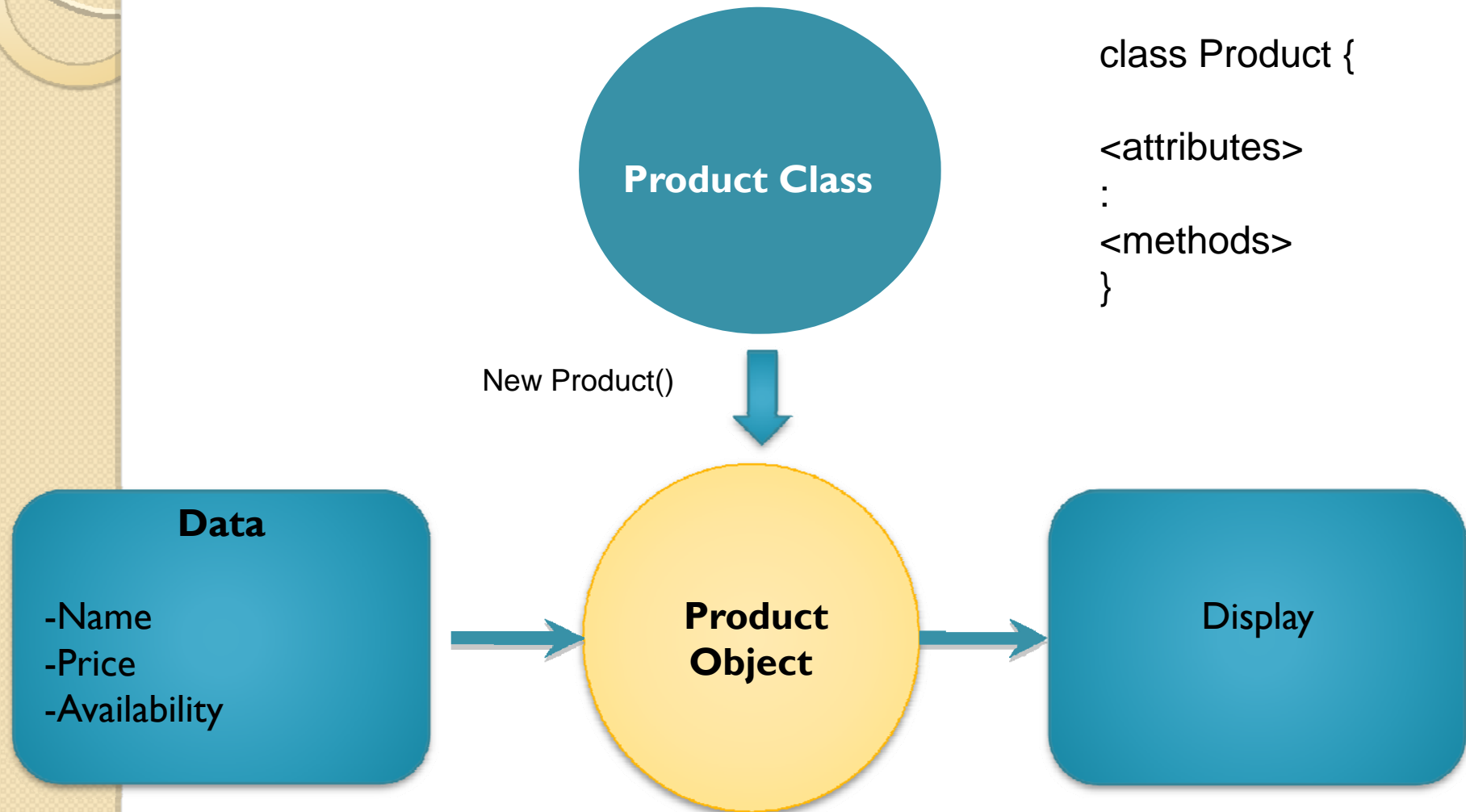
Output Step 2



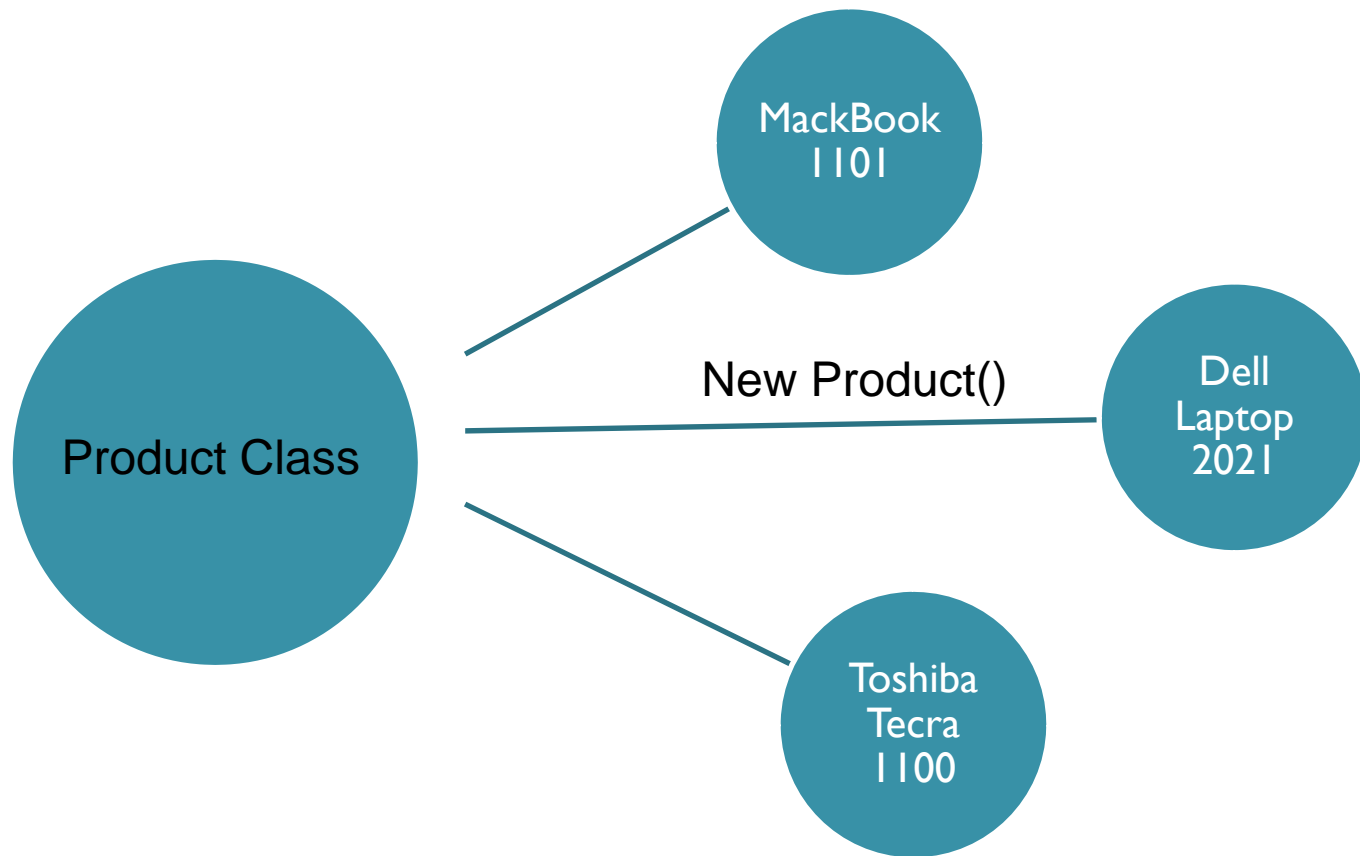
How to create and move data in and out of objects?



The approach:

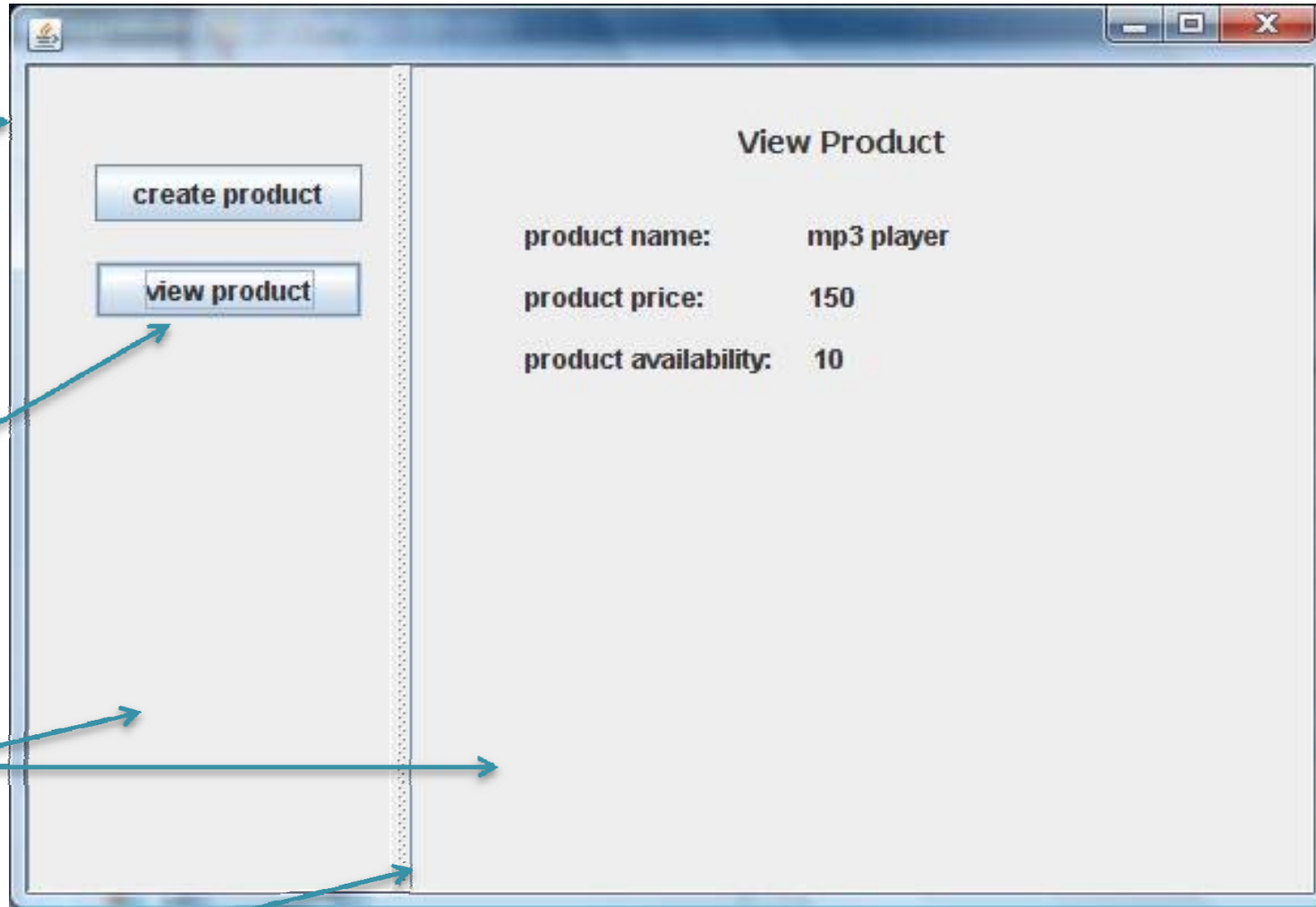


Instantiation: From classes to objects



Swing application structure

JFrame



JButton

JPanel

JSplitPane

Pass the product object

Jframe (MainJFrame)

**Product
Object**

Customer

Create Product JPanel

**Product
Object**

Input fields

NameTextField

PriceTextField

Jframe (MainJFrame)

**Product
Object**

Customer

Create Product JPanel

Input fields

NameTextField

PriceTextField

name
price

Putting the display panels together

Jframe (MainJFrame)

**Product
Object**

**Product
Object**

Display Product JPanel

Output fields

NameTextField

PriceTextField

Putting the display panels together

Jframe (MainJFrame)

Display Product JPanel

**Product
Object**

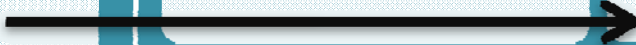
Output fields

name



NameTextField

price



PriceTextField

Procedure

1. Create a new project
2. Define Business Package
 - Create product class
3. Define UserInterface Package
 - Define JFrame
 - Define Jpanels (Create and View)
 - Button listeners

Create Product Class

under the business package

Attribute

Product

- name
- price
- availability
- description

Method

Product Class

```
public class Product {  
  
    private String name;  
    private String price;  
    :  
  
    public String getName()    // retrieve data  
    {    return name;    }  
    :  
  
    public void setName(String n)    // keep data  
    {    name = n;    }  
    :  
}
```


Define MainJFrame Class

under the userinterface package

Create a global variable of product for this class

```
class MainJFrame {
```

Constructor

```
    public MainJFrame() {
```

```
        initComponents();
```

```
    }
```

```
}
```

Method that creates visual components

Define MainJFrame Class

under the userinterface package

Create a global variable of product for this class

```
class MainJFrame {  
    private Product product; // Global Variable  
  
    public MainJFrame() {  
  
        initComponents();  
        product = new Product(); // Instantiate the object  
        (global variable)  
    }  
}
```

Define CreateProductJPanel class

under the userinterface package

```
class CreateProductJPanel () {
```

Constructor

```
    CreateProductJPanel () {  
        initComponents();  
    }
```

```
<other stuff>
```

```
}
```

Method that creates visual components

Define CreateProductJPanel class

under the userinterface package

```
class CreateProductJPanel() {
```

```
CreateProductJPanel(Product p) {  
    initComponents();  
    product = p;  
    nameTextField.setText(product.getName());  
    priceTextField.setText(product.getPrice());  
}
```

```
<other stuff>  
}
```

Constructor

Create Product

- In the constructor of **CreateProductJPanel** class

```
public CreateProductJPanel(Product p) {  
    initComponents();  
  
}
```

View Product

- In the constructor of **ViewProductJPanel** class

```
public ViewProductJPanel(Product p) {  
    initComponents();  
    this.product = p;  
  
    nameTextField.setText(product.getName()  
));  
  
    priceTextField.setText(product.getPrice()  
));  
  
}
```

Button Events

When Create or View buttons are clicked on the left side, the following actions should be performed respectively.

Create Button

```
CreateProductJPanel j = new  
    CreateProductJPanel (product) ;  
jSplitPanel1.setRightComponent (j) ;
```

View Button

```
ViewProductJPanel v = new  
    ViewProductJPanel (product) ;  
jSplitPanel1.setRightComponent (v) ;
```

Create Button

When “create button” is clicked, following actions should be performed in the action perform method of the button

```
p.setName(nameField.getText()); //  
p.setPrice(priceField.getText());  
p.setAvailability(availabilityField.getText());
```


View Button

When “create button” is clicked, following actions should be performed in the action perform method of the button

```
nameTextField.setText(p.getName());  
priceTextField.setText(product.getPrice());
```