CT2106
Object Oriented Programming

Dr. Frank Glavin
Room 404, IT Building
Frank.Glavin@University*of*Galway.ie
School of Computer Science

University *of* Galway.ie

Last Week

- Much of OOP is about making good modeling decisions
- A model is a simplified representation of reality
- Core modeling decisions: what are the objects, what are their responsibilities, what are their associations with each other
- Start by identifying the objects and relationships in the problem domain – these are candidate objects
- It is important to set your code an objective or test before writing the code
- Create the stub code for your classes
- Development, particularly OO development is **incremental** and **iterative**

Ollscoil na Gaillimhe
University of Galway

This lecture

This lecture will prepare the groundwork for the next major topic we cover in OOP:

- **Inheritance**

Today's topics:

- Object equivalence

Ollscoil na Gaillimhe
University of Galway

- Open BlueJ
- Create a new Project
- Make sure Code Pad is displayed
- (View-> Show Code Pad)

Ollscoil na Gaillimhe
University of Galway

Instructions 1

1. Create a String variable **str1** to hold a String value “Java”
2. Type **str1** into CodePad. It should return the value “Java”
3. Create another String variable **str2** to hold a String value “Ja”
4. Create another String variable **str3** to hold a String value “va”
5. Create another String object **str4** to hold the String value when **str3** is added to **str2**
6. Type **str4** into CodePad. It should return the value “Java”

Ollscoil na Gaillimhe
University of Galway

Instructions 2

You are now going to check for the equality of the values of **str1** and **str4**

1. Write an **if** statement to test if **str1** has the same value as **str4**
2. The if statement should print out **true** if **str1** has the same value as **str4** and **false** if they do not print out the same value

(Hold down the Shift and Enter keys to enter more than one line in CodePad)

Ollscoil na Gaillimhe
University of Galway

Hint

```
int x = 8;
int y = 9;

if(x==y){
    System.out.println(“true”);
} else{
    System.out.println(“false”);
}
```

Ollscoil na Gaillimhe
University of Galway

How many wrote something like this?

```
String str1 ="Java";
String str2 ="Ja";
String str3 = "va";
String str4 = str2+str3;

if(str1==str4){
    System.out.println("true");
} else{
    System.out.println("false");
}
```

Ollscoil na Gaillimhe
University of Galway

What will the output be?

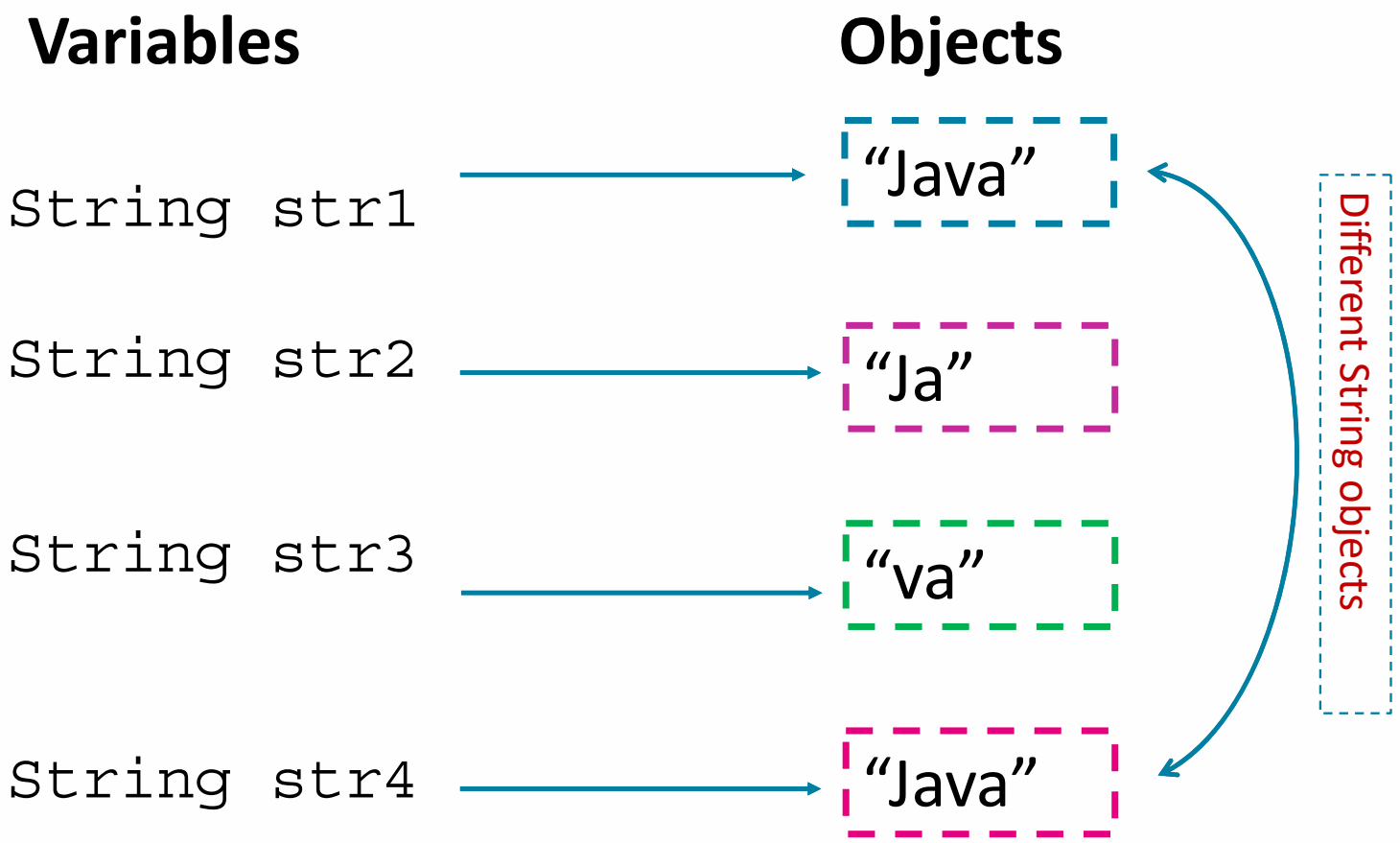
```
if(str1==str4){
    System.out.println("true");
} else{
    System.out.println("false");
}
```

Ollscoil na Gaillimhe
University of Galway

Why?

- Why is the value of **str1** not equal to the value of **str4**
- The answer is that the values of **str1** and **str4** are memory references to **different objects**
- It doesn't t matter that the objects may contain the same data ("Java")
- When you use **==** with reference variables **you are simply checking if the variables point to the same object**

Ollscoil na Gaillimhe
University of Galway


Variables
Objects
String str1
String str2
String str3
String str4
“Java”
“Ja”
“va”
“Java”
Different String objects

```
if(str1==str4){
    System.out.println("true");
} else{
    System.out.println("false");
}
```

The value of **str1** is the memory location where its String object is stored
The value of **str4** is the memory location where its String object is stored
So **str1** is not equal (==) to str4

Ollscoil na Gaillimhe
University of Galway

Object Equality

- When checking for equality between objects you must use the **equals** method
- **The equals method is an instance method that all objects have**
- Its specific purpose is to define equality between objects
- It returns a **boolean** value

Ollscoil na Gaillimhe
University of Galway

You can download this code snippet from Blackboard

StringEqualityDemo

Compile | Undo | Cut | Copy | Paste | Find... | Close | Source Code

```
 */
public class StringEqualityDemo
{

    /**
     * main method used to illustrate String equality
     *
     */
    public static void main (String[] args)
    {
        String str1 ="Java";
        String str2 ="Ja";
        String str3 = "va";
        String str4 = str2+str3;

        if(str1==str4){
            System.out.println("true");
        } else{
            System.out.println("false");
        }


    }
}
```

Ollscoil na Gaillimhe
University of Galway

Rewrite the code and run

```
String str1 ="Java";
String str2 ="Ja";
String str3 = "va";
String str4 = str2+str3;

if(str1.equals(str4)){
    System.out.println("true");
} else{
    System.out.println("false");
}
```

Output:

```
true
```

Ollscoil na Gaillimhe
University of Galway

In this case, we use the **equals** method of the String object referenced by **str1**

It accepts the value of **str4** as an input parameter and returns true or false

```
if(str1.equals(str4)){
    System.out.println("true");
} else{
    System.out.println("false");
}
```

Ollscoil na Gaillimhe
University of Galway

equals must be **commutative**

```
str1.equals(str4)
```

must return the same boolean value as…

```
str4.equals(str1)
```

Ollscoil na Gaillimhe
University of Galway

Every object has an equals method

- **Every single object has an equals method**
- Because evaluating the equality between objects is a very common function
  - E.g for searching, sorting
- For the built-in classes of Java, the equals method will already be defined
- But for any class that you define **you will have to write the equals method**

Ollscoil na Gaillimhe
University of Galway

Tutorial - Collections

- We will now spend a few minutes looking at the collection tutorial
  - There are two separate PDFs that can be found in Week 4 on Blackboard
- We will also look at looping over items in a collection

Ollscoil na Gaillimhe
University of Galway