Review Session

Classes, Objects, String Formatting

Colorado State University
Department of Computer Science

Slides Originally Created by Albert Lionelle (Albert.Lionelle@colostate.edu)

Classes and Objects

- If we want to create an object (instance of a class) we need to use the class constructor (new) – we will talk and learn more about this beginning next week.
  - If you are wondering, String is a special kind of class that does not use new explicitly to create an object ☺
- Consider the following instruction:

```java
Scanner  scanner = new Scanner(System.in);
```

- Since Java needs to relate a type to every variable that is used, in the previous instruction we are using the type of a class, named Scanner, to create a variable, in this case an object (instance of a class) named scanner.
  We use the constructor (new) and the name of the class (that is actually the name of the constructor).
  Scanner needs a parameter to create an object, so we provide System.in, meaning that we are going to read data from the console.

Classes and Objects

- Now consider the following instruction:

```
        Rectangle example =  new Rectangle(10, 4);
```

- Rectangle is a class that we created during our lecture. The instruction above is creating an object (instance of a class) named example. The type of that object is Rectangle.
- We are using "new" to call the constructor Rectangle, which does not has any parameter.
- Below we have two uses of the example object. Remember: to call a method of an object you need to have *objectName.methodName(list of parameters)*. You also need to pay attention to what the method is going to return.

```
public static void main(String[] args) {
     Rectangle example =  new Rectangle(10, 4);
     double width = example.getWidth();
     System.out.printf("Width: %d\n", width);
     System.out.println(example.getHeight());
 }
```

```
public class Rectangle {

  private double height;
  private double width;

  public Rectangle(double height, double width){
    this.height = height;
    this.width = width;
  }

  public void setHeight(double height){
      this.height = height ;
  }

  public double getHeight(){
    return height;
  }

  public double getWidth(){
    return width;
  }
}
```

String formatting

- In the example we are printing 3 different variables in the same printf
- The order of the variables after the comma matters
- Everything between “ ” will be printed, including commas, spaces, etc.
- %d – will be substitute by value1, which is int
- %.2f – will be substitute by value2, which is a double, and it will print just 2 decimals after the point
- %% - will print one %
- \n – will do a new line
- %s – will be substitute by color, which is a String

```java
public class StringFormatProgram {
   public static void main(String[] args) {
      int value1 = 12;
      double value2 = 40.213;
      String color = “blue";
      System.out.printf(“%d, %.2f%% \n %s”, value1, value2, color);
     }
}
```

```
12, 40.21%
 blue
```