

# CSCI 2120:

## Software Design & Development II

*UNIT4: UI management*

*GUI framework*

**JavaFX Text Attribute: Font**

# Overview

1. Introduction
2. Create Font Instance
3. Using the Font Instance
4. List Installed Font Families and Names

# Introduction

- The JavaFX **Font** class, `javafx.scene.text.Font`, enables you to load different Fonts for use in your JavaFX applications.
- A **font** is a **text style**. All text rendered with the same font will look similar.
- In this JavaFX **Font** lecture you will learn how to **load** and **set fonts** in JavaFX.

# Create Font Instance

To use **fonts** in JavaFX you must create a JavaFX **Font** instance. The easiest way to create a JavaFX **Font** instance is to use the **static factory methods** in the **Font** class.

The following example shows how to create JavaFX **Font** instances using the many variations of the **Font** class **static factory methods**:

```
String    fontFamily = "Arial";
double    fontSize   = 13;
FontWeight fontWeight = FontWeight.BOLD;
FontPosture fontPosture = FontPosture.ITALIC;

Font font1 = Font.font(fontFamily);
Font font2 = Font.font(fontSize);
Font font3 = Font.font(fontFamily, fontSize);
Font font4 = Font.font(fontFamily, fontWeight, fontSize);
Font font5 = Font.font(fontFamily, fontPosture, fontSize);
Font font6 = Font.font(fontFamily, fontWeight, fontPosture, fontSize);
```

As you can see, the **Font** factory methods enable you to create **Font** instances representing different **font families, font sizes, font weights and font postures**.

# Using the Font Instance

Once you have created a JavaFX **Font** instance you use it by setting it on whatever JavaFX component capable of using a **Font**. For instance, you can set it on a **JavaFX Text** control.

Here is an example of setting a **Font** instance on a **Text** element:

```
Font font = Font.font("Arial");  
  
Text text = new Text("This is the text");  
text.setFont(font);
```

Precisely how a **Font** object is applied to a given JavaFX element depends on the specific JavaFX element. In the JavaFX **Text** example the **Font** object is applied via the **Text setFont()** method, as shown above.

# List Installed Font Families and Names

The JavaFX `Font` class provides **two static methods** that can **list the font families and font names** installed on the system. These methods are the `getFamilies()` and `getFontNames()` methods.

Here are some examples of calling these methods:

```
List<String> fontFamilies = Font.getFamilies();  
List<String> fontNames   = Font.getFontNames();
```

Both **font family names** and **font names** can be used when creating a `Font` instance. Provide either the font family name or font name in the `Font` factory method `fontFamily` parameter.

To see the actual names, loop through the lists above and print out their names, like this:

```
for(String item : fontFamilies) {  
    System.out.println(item);  
}  
  
for(String item : fontNames) {  
    System.out.println(item);  
}
```

END