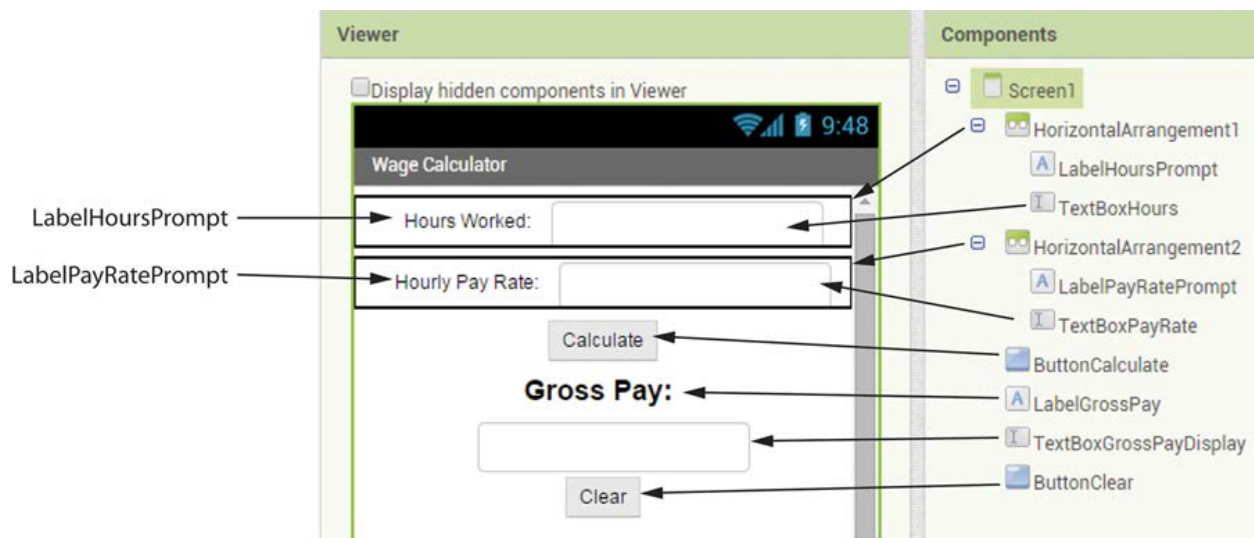


Tutorial 4-3: Creating the Wages App

At a business, if an employee works more than 40 hours in a week, it is said that the employee has worked overtime. For example, an employee that has worked 45 hours in a week has worked = overtime hours. Employees that work overtime get paid their regular hourly pay rate for the first 40 hours plus 1.5 times their regular hourly pay rate for all hours over 40.

In this tutorial, you will create an app that calculates an employee's gross pay, including overtime pay. The app allows the user to enter the number of hours worked and the hourly pay rate into text boxes. Figure 4-38 shows the app's screen in the Viewer, along with the names of the components.

Figure 4-38 The App's User Interface



When the user clicks the ButtonCalculate button, the gross pay is calculated in the following manner:

If the hours worked is greater than 40:

- regular pay = hourly pay rate \times 40
- overtime hours = hours worked - 40
- overtime pay = overtime hours \times (hourly pay rate \times 1.5)
- gross pay = regular pay + overtime pay

Else:

- gross pay = hours worked \times hourly pay rate

As you can see, there are two very different calculations, depending on whether the user has worked more than 40 hours. Regardless of which calculation is used, the result is the

gross pay. In this app, we will use an if then else block to determine whether the user has worked overtime and perform the correct calculation.

Step 1: Start a new project named Wages.

Step 2: Set up the app's screen with the components shown in Figure 4-38. Refer to Table 4-3 for the relevant property settings for each component.

Table 4-3 Component property settings

Component	Relevant Property Settings
Screen1	AlignHorizontal = Center
	Title = Wage Calculator
HorizontalArrangement1	AlignHorizontal = Center
	Width = Fill parent
	Height = Automatic
LabelHoursPrompt	Text = Hours Worked:
TextBoxHours	Enabled = checked
	NumbersOnly = checked
	Hint = Hours Worked
	Width = Automatic
	Height = Automatic
HorizontalArrangement2	AlignHorizontal = Center
	Width = Fill parent
	Height = Automatic
LabelPayRatePrompt	Text = Hourly Pay Rate:
TextBoxPayRate	Enabled = checked
	NumbersOnly = checked
	Hint = Hourly Pay Rate
	Width = Automatic
	Height = Automatic
ButtonCalculate	Text = Calculate
LabelGrossPay	FontBold = checked
	FontSize = 20
	Text = Gross Pay:
	TextAlignment = Center

TextBoxGrossPayDisplay	Enabled = unchecked
	FontBold = checked
	FontSize = 20
	Hint 5 cleared
	TextAlignment = Center
	Width = Automatic
	Height = Automatic
ButtonClear	Text = Clear

Step 3: Now you will program the blocks for the app. Open the Blocks Editor.

Step 4: Create a Click event handler for the ButtonCalculate button.

Step 5: Now you will create and initialize the local variables needed to calculate the gross pay. Create an initialize local name to block and, as shown in Figure 4-41, insert it into the ButtonCalculate.Click event handler.

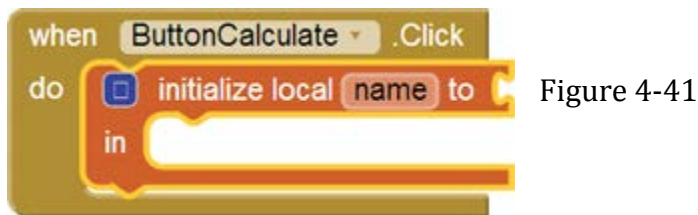


Figure 4-41

Step 6: Use the variable initialization block's mutator bubble to add three more variable names to the block. (The initialization block should have a total of four variables.) Change the variable names to those shown in Figure 4-42. Initialize each variable to the value 0.



Figure 4-42

Step 7: Create an if then block and use its mutator bubble to change it to an if then else block. Insert the if then else block into the variable initialization block, as shown in Figure 4-43.

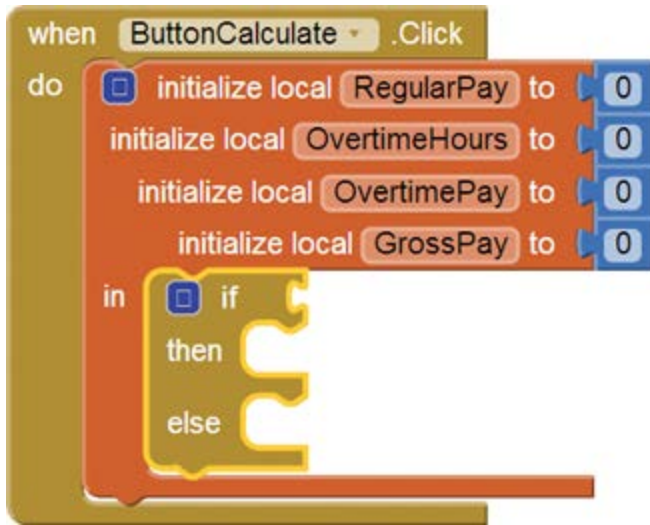


Figure 4-43

Step 8: Insert the Boolean expression shown in Figure 4-44 into the if socket of the if then else block.



Figure 4-44

Step 9: The if then else block determines whether the hours worked is greater than 40. If that is true, the blocks in the then section will calculate the gross pay with overtime and assign the result to the GrossPay variable. Add the blocks shown in Figure 4-45 to the if then else block's then section.

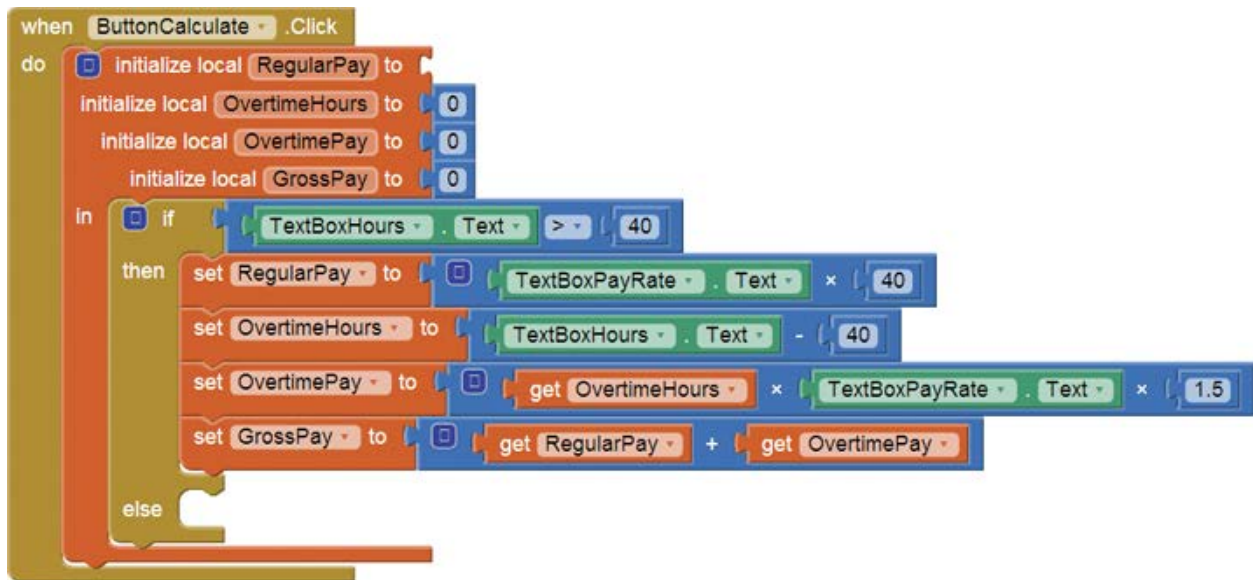
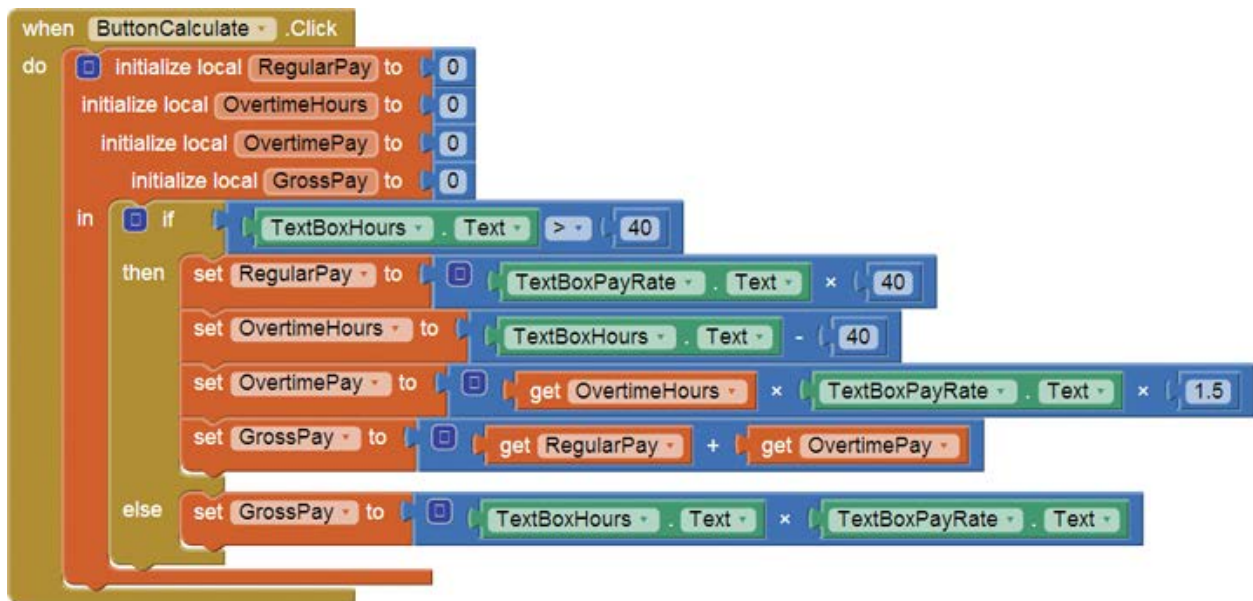


Figure 4-45



Step 10: If the hours worked is not greater than 40, the if then else block's else section will calculate the gross pay without overtime and assign the result to the GrossPay variable. Add the blocks shown in Figure 4-46 to the if then else block's else section.

Figure 4-46

Step 11: The last action to be performed by the ButtonCalculate component is to display the gross pay. Add the blocks shown in Figure 4-47.

```

when ButtonCalculate . Click
do
  initialize local RegularPay to 0
  initialize local OvertimeHours to 0
  initialize local OvertimePay to 0
  initialize local GrossPay to 0
  in
    if TextBoxHours . Text > 40
    then
      set RegularPay to TextBoxPayRate . Text × 40
      set OvertimeHours to TextBoxHours . Text - 40
      set OvertimePay to get OvertimeHours × TextBoxPayRate . Text × 1.5
      set GrossPay to get RegularPay + get OvertimePay
    else
      set GrossPay to TextBoxHours . Text × TextBoxPayRate . Text
    set TextBoxGrossPayDisplay . Text to format as decimal number get GrossPay
    places 2

```

Add these blocks in Step 11.

Step 12: The ButtonClear button will simply clear the Text property of the TextBox components. Create the button's Click event handler as shown in Figure 4-48.

```

when ButtonClear . Click
do
  set TextBoxHours . Text to ""
  set TextBoxPayRate . Text to ""
  set TextBoxGrossPayDisplay . Text to ""

```

Figure 4-48

Step 13: Test the app in the emulator or on your device. First enter 40 for the number of hours worked and 20 for the hourly pay rate. Click the Calculate Gross Pay button and the app should display 800.00 as the gross pay. No overtime hours were worked, so the gross pay is simply calculated as hours worked × hourly pay rate.

Next, click the Clear button. The text boxes should clear. Enter 50 for the number of hours worked and 20 for the hourly pay rate. Click the Calculate button and the app should display 1100 as the gross pay. This time, more than 40 hours were worked, so the app calculated the gross pay to include overtime pay. Continue to test the app as you wish.