BCS 371 Lab – Kotlin Lambdas

Overview

Create an app that uses Kotlin lambdas in various ways. The Logcat will be used for output.

Create a project

Create a new Android application in Android Studio. Choose the **Empty Activity** type to create an empty activity that uses Jetpack Compose.

Cleanup project start code and run app

Do the following in MainActivity.kt:

- Remove the Greeting and GreetingPreview functions from the MainActivity class.
- Remove all code from the onCreate function except for the call to super.onCreate.
- Add a println statement with the message “Hello Lambda” (it will print in the Logcat window).
- Run the app to make sure it works.

Write message function

Write a function named message that takes a string as a parameter. It should print the parameter in the Logcat window. Call message from MainActivity.onCreate.

Variable storing a function reference

Create a variable and store a reference to the message function in it. Call message using the variable. Do this all in MainActivity.onCreate.

Variable with lambda expression

Do the following in MainActivity.onCreate.

Create a variable that contains a lambda expression that will apply a raise to a salary (creating an anonymous function). The function should have two Double parameters and return a Double. The parameters are raise and salary. Raise will be a decimal (.1 for 10%, .2 for 20%). The function should calculate the raise amount and add that number to salary to calculate a new salary. It should return the new salary.

Call the function using the variable and show the new salary in the Logcat window.

Pass function reference as a parameter

Add a new member function named useFunctionRefParm to MainActivity.

- The useFunctionRefParm function should have three parameters.
  - First parameter. A function reference with the format: (Double, Double)->Double
  - Second parameter. A Double to hold the salary.
  - Third parameter. A Double to hold the raise percentage.
- Inside the body of useFunctionRefParm, call the first parameter (a function reference), passing in raise and raise percentage. Display the value it returns in the Logcat window.

In MainActivity.onCreate, call useFunctionRefParm, passing in the raise function (that you defined in the previous section), a salary value, and raise percentage value.

Create and use SalaryList class

Create a class named SalaryList.

- Declare a member variable named salaries with the type List<Double>.
- Write an init block.
  - Initialize salaries list with the following values: 80.0, 100.0, 90.0, 70.0, 120.0
- Write a function show. It should have no parameters or return type. It should print all salary values from the list in the Logcat window.
- Write a function showWithRaise. It should take a function reference as a parameter (we will be passing in the raise function as a parameter). The function parameter should match the previously defined raise function (two Double parameters and a Double return type). Call the function parameter on all items in salaries and print the result of applying the function in the Logcat window. Pass in .1 for the raisePct.
- In MainActivity.onCreate do the following:
  - Create an instance of SalaryList.
  - Call show on the instance.
  - Call showWithRaise on the instance (pass in the raise function as a parameter).
- Run the app and check the output.

Update SalaryList class

Make the following updates to SalaryList:

- Change the class header so that it uses a primary constructor that takes a function reference as a parameter. The function reference parameter should be for functions that take two Double parameters and return a Double. This parameter will serve as a new member variable in the class.
- Update the showWithRaise function so that it does not take any parameters. It should now apply the primary constructor function parameter to all items in salaries.

- Do the following in MainActivity.onCreate:
    - Pass the raise function into the SalaryList constructor.
    - Update the showWithRaise function call so that it takes no parameters.
- Run the app.