Congratulations! You passed!
Grade received 100%Latest Submission Grade 100%To pass 80% or higher

Go to next item

1. Which of these are correct ways to instantiate a function type. Select all that apply. 1 / 1 point

- [ ] Using the function name.
- [x] Using a callable reference to existing declaration using the '::' operator

Correct
Correct! You can use a callable reference such as '::functionName'.

- [x] Using a lambda expression.

Correct
Correct! You can use a lambda expression to instantiate a function type.

- [x] Using instance of a user defined class that implements a function type as an interface.

Correct
Correct! You can instantiate a function using a defined class that implements a function type as an interface.

2. Which of these is a correct usage of lambda expression for the given function definition? 1 / 1 point

```
1   fun execute(number: Int, function: (Int) -> String) {
2    println(function(number))
3   }
```

- ( ) **execute(100) ("Score $it")**
- (•) **execute(100) { "Score $it" }**
- ( ) **execute("Score") { "$it 100" }**

Correct
Correct! The above function takes in an 'Int' argument and then prints a string by concatenating it with the 'Score'.

3. Which of the these is a correct lambda expression syntax? 1 / 1 point

- ( ) **val difference: (Int, Int) -> Int = x: Int, y: Int -> x - y**
- (•) **val difference: (Int, Int) -> Int = { x: Int, y: Int -> x - y }**
- ( ) **val difference: (Int, Int) -> Int = x: Int, y: Int -> { x - y }**

Correct
Correct! This is the correct syntax

4. Which listener interface provided by the Android framework is used to listen for a button press event? 1 / 1 point

- (•) View.OnClickListener
- ( ) View.OnPressListener
- ( ) View.OnTapListener

Correct
Correct! The 'View' class contains an interface 'OnClickListener' that has a method 'onClick' which gets called on events such as a button press.

5. Which of these are higher-order functions? Select all that apply. 1 / 1 point

- [ ] **fun display(x: (Int)) -> Unit**
- [x] **fun display(x: (Int) -> Unit)**

Correct
Correct! This is a higher-order function as it takes another function as a parameter.

- [x] **fun display(): (Int) -> Unit**

Correct
Correct! This is a higher-order function as it returns a function.

- [ ] **fun display(x: Int) : Unit**

6. What is the output of this code? 1 / 1 point

```
1   val number = 3
2   var output = 2
3   repeat(5) { index ->
4    output += (index * number)
5   }
6   println(output)
7
```

- (•) 32
- ( ) 30
- ( ) 47

Correct
Correct! You correctly calculated the output of the given code.

7. What is the output of the following code? 1 / 1 point

```
1   var sum = 0
2
3   val numberList = listOf(1, 4, 6, 7, 9)
4   numberList.forEach { number ->
5    sum += number
6   }
7   println(sum)
```

- ( ) 9
- (•) 27
- ( ) 1

Correct
Correct! The above code would iterate over each element and then add each element's value to variable named 'sum'.

8. What is the output of this code: 1 / 1 point

```
1   data class Chocolate(
2    val flavor: String,
3    val price: Int
4   )
5   val list = listOf(
6    Chocolate("Dark", 7),
7    Chocolate("Milk", 4),
8    Chocolate("Coffee", 2)
9   )
10  val output = list.map {
11   it.flavor
12  }
13  println(output)
14
```

- ( ) [7, 4, 2]
- ( ) [Chocolate(flavor=Dark, price=7), Chocolate(flavor=Milk, price=4), Chocolate(flavor=Coffee, price=2)]
- (•) [Dark, Milk, Coffee]

Correct
Correct! The above code transforms the initial list to a new list that contains values of 'flavor'.

9. What is the output of this code: 1 / 1 point

```
1   data class Chocolate(
2    val flavor: String,
3    val price: Int
4   )
5   val list = listOf(
6    Chocolate("Dark", 7),
7    Chocolate("Milk", 4),
8    Chocolate("Coffee", 2)
9   )
10  val output = list.filter {
11   it.price > 3
12  }
13  println(output)
14
```

- (•) [Chocolate(flavor=Dark, price=7), Chocolate(flavor=Milk, price=4)]
- ( ) [Chocolate(flavor=Coffee, price=2)]
- ( ) [Chocolate(flavor=Dark, price=7), Chocolate(flavor=Milk, price=4), Chocolate(flavor=Coffee, price=2)]

Correct
Correct! The code above filters the 'chocolate' elements that have 'price' > 3, and returns a new list with only those elements that comply to the condition.

10. What is the output of this code: 1 / 1 point

```
1   val list = listOf(9, 3, 1, 6)
2   val output = list.fold(1) { x, y ->
3    x + y
4   }
5   println(output)
6
```

- ( ) 18
- (•) 20
- ( ) 1

Correct
Correct! The fold function accumulates a value starting from the initial value of '1' and then applies the operation to each element in the list.