ÆON Credit Service Systems (Philippines) Inc.

Oracle Associate Level Certification
Java SE 8 Specific Exam Introduction

Prepared by: Alexis John Cantiga

OBJECTIVES

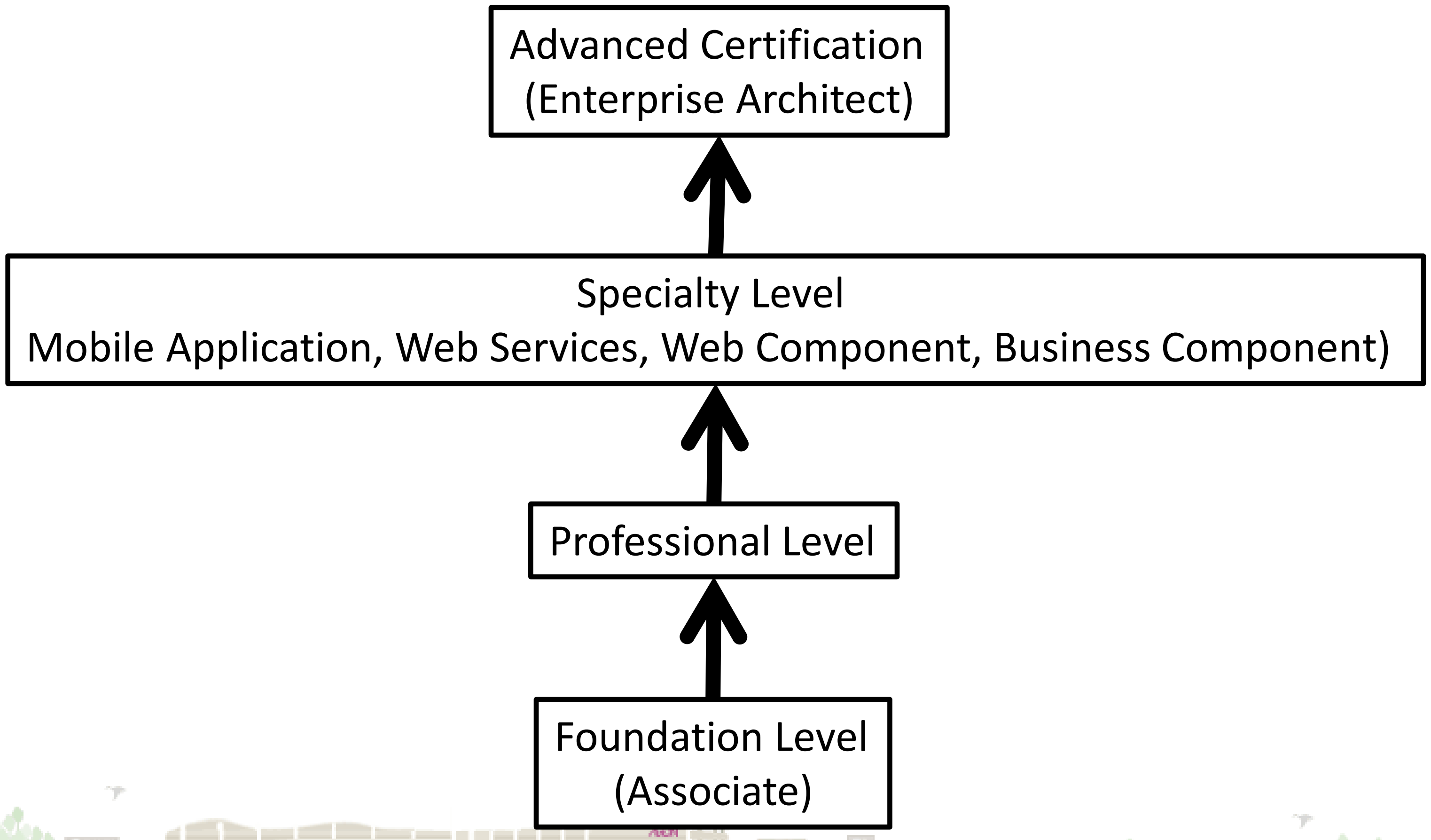
This session aims to:

- Introduce the new basic features of Java SE 8 included in the Associate Level Certification
- Identify the correct syntax of a Lambda expression
- Introduce LocalDate, LocalTime, and LocalDateTime
- Differentiate class `default` access from interface `default` method

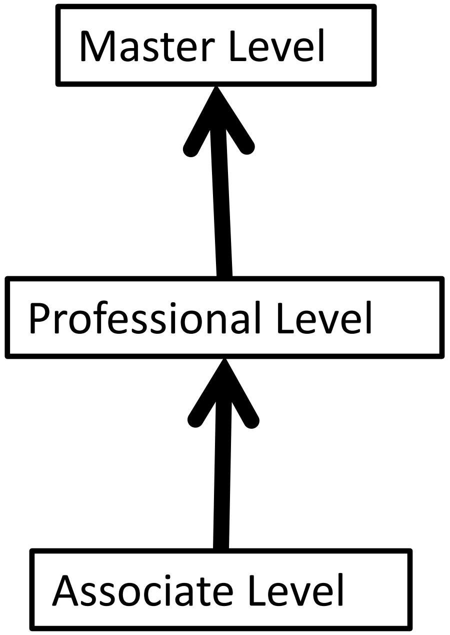
OUTLINE

- Brief History of Java Certifications
- Syntax of Lambda Expressions
- Introduction to LocalDate, LocalTime, and LocalDateTime
- Interface Default methods

SUN JAVA CERTIFICATION PATH (PREVIOUS)



ORACLE JAVA CERTIFICATION PATH (CURRENT)



EXAM COVERAGE

- Java Basics
- Working with Java Data Types
- Using Operators and Decision Constructs
- Creating and Using Arrays
- Using Loop Constructs
- Working with Methods and Encapsulation
- Working with Inheritance
- Handling Exceptions
- Working with Selected classes from the Java API

LAMBDA EXPRESSION

Long Format

```
(String str) -> {return "1".equals(str);}
```

Short-hand Format

```
str -> "1".equals(str)
```

LAMBDA EXPRESSION WITH PREDICATE<T> INTERFACE

- standard functional interfaces included in JDK
- Predicate<T> has only one method that returns a boolean

```
interface Predicate<T> {
        boolean test(T t);
}
```

LAMBDA EXPRESSION WITH PREDICATE<T> INTERFACE

```
private int countStringsContainsValue (List<String> textList,
                        Predicate<String> checker) {
            int counter=0;
            for(String text : textList) {
                        if(checker.test()) {
                                    counter++;
                        }
            }
            return counter;
}
```

LAMBDA EXPRESSION WITH PREDICATE<T> INTERFACE

Long Format

countStringsContainsValue(textList, (String str) -> {returns str.contains("8"); });

Short-hand Format

countStringsContainsValue(textList, str -> str.contains("8"));

JAVA SE 8 DATE AND TIME

-From `java.util` package

```
Date now = new Date();
System.out.println(now); // Tue Jan 01 01:02:03 CST 2016
```

JAVA SE 8 DATE AND TIME

Most Common Usage

```
LocalDateTime now = LocalDateTime.now();     // The current date and time
LocalDate.of(2012, Month.DECEMBER, 12); // from values
LocalTime.of(17, 18); // Hour and minute
LocalTime.parse("10:15:30"); // From a String
```

Getting the specific part of a LocalTime object:

```
LocalTime specificPartOfTime = time.truncatedTo(ChronoUnit.MINUTES);
```

JAVA SE 8 DATE AND TIME

```
// Instantiation
Period period = Period.of(3, 2, 1); // 3 years, 2 months, 1 day

// Example usage
LocalDate date = LocalDate.now();
date = date.plus(period);
```

DEFAULT METHODS VS DEFAULT ACCESS MODIFIER

Access modifiers:
- public
- protected
- default (no modifier specified)
- private

Access modifiers determines whether other classes can use a particular field or Invoke a particular method.

DEFAULT METHODS VS DEFAULT ACCESS MODIFIER

Default methods enable you to add new functionality to the interfaces of your ibraries and ensure binary compatibility with code written for older versions of those interfaces.

DEFAULT METHODS

```
public interface Display {
        public abstract void enableFullResolution(int width, int height);
        public abstract void enable3dDisplay(int width, int height, int depth);
        public default void enableHologram(){
                // added for future implementation
        }
}
```

DEFAULT METHODS

```
// Monitor.java
public class Monitor implements Display {
        public void enableFullResolution(int width, int height) {
                // TODO: code implementation
        }
        public void enable3dDisplay(int width, int height, int depth) {
                // TODO: code implementation
        }
}
```

DEFAULT METHODS

```
// HoloWatch.java
public class HoloWatch implements Display {
        public void enableFullResolution(int width, int height) {
                // TODO: code implementation
        }
        public void enable3dDisplay(int width, int height, int depth) {
                // TODO: code implementation
        }
        public void enableHologram() {
                // TODO: code implementation
        }
}
```

QUESTIONS?