CO225: Software Construction
Tutorial 2

Answer the following questions. These are typical questions asked during interviews.

1. Can you have static methods in an interface deceleration? Explain.

    - Yes
    - From Java 8 onwards we are allowed to define static methods in interfaces. Static methods are defined in the interface using the static keyword. These methods contain the complete definition of the function. Static methods cannot be overridden or changed in the implementation class. This feature helps us in providing security and in avoiding undesired results caused in case of poor implementation in the implementation class. To use a static method, interface name should be instantiated with it.
    - But before Java 8 version we couldn’t use static methods in interfaces. We could only have public abstract methods in interfaces.

2. Can you have private methods in a interface declaration? Explain.

    - Yes
    - From Java 9 onwards we are allowed to define private methods in interfaces. The private modifier is used to define these methods and no lesser accessibility than private modifier. The private methods cannot be abstract. These methods can be only used inside an interface. Private static methods can be used inside other static and non static methods but private non static methods cannot be used inside private static methods. These private methods will improve code re-usability inside the interface and will provide choice to expose only our intended methods implementations to users.
    - But before Java 9 version we couldn’t use private methods in interfaces. We could only have public abstract methods in interfaces before Java 8 version.

3. Can you extend an interface? Why not?

    - A class cannot extend an interface
        - ♦ By extending an interface we could inherit all the fields and methods in that interface. But an interface contains unimplemented methods. Therefore a class cannot extend an interface.
    - An interface can extend an interface

4. You want to give a partial implementation of a class to which a developer has to add more functionalities before making objects. Also you want to make sure that the developer will not be able to modify any of the methods that you have written. How would you achieve this?

    - Defining the methods in the base class using the private modifier.
    - Defining the methods in the base class using static keyword.
    - Defining the methods in the base class using final keyword.

5. Java does not allow multiple inheritance. Find a language which does allow multiple inheritance and explain why Java opted not to have it.

    - C++ allows multiple inheritance.

- Java doesn’t allow multiple inheritance to avoid the ambiguity caused by it. One of the example of such problem is the diamond problem that occurs in multiple inheritance.

6. When class B inherit from class A, does B inherit the A's constructor?

- No
- Reasons
  - The constructor should have the same name as the class name. Therefore if constructors are inherited in the child class then the child class would have the parent class constructor with the parent class name.
  - It will be impossible to achieve encapsulation.
  - The constructor is called when the object of the class is created. Therefore if constructors are inherited there will be no point of creating child class object using parent class constructor notation. Ex -: B b = new A( );
  - super( ) can be used inside the child class to call the parent class constructor.

7. What is a constructor and can you have a class without an constructor? Explain.

- Constructor
  - A constructor is a special type of method which is used to initialize the object. At least one constructor is called when an object is created using new( ) keyword. We can use constructors to give initial values to the instance variables defined by the class or to perform any other start-up procedures required to create a fully formed object.
- No. All classes have constructors.
  - If you do not define a constructor explicitly a default constructor is automatically provided by Java, which initializes all member variables to zero.

8. If there are no command-line arguments provided to the program, will the String array provide to main method be null or empty?

- It is empty.
  - If we print args.length it will print zero. Therefore the array is empty.
  - But if the array was supposed to be null, then on attempting to print args.length it would have thrown a NullPointerException.

9. What is a JIT compiler (*not covered in lectures. Search the web for answers*)

- The Just-In-Time (JIT) compiler is a component of the JRE (Java Runtime Environment) that improves the performance of Java applications at run time. It helps improve the performance of Java programs by compiling bytecodes into native machine code at runtime .

10. Can an abstract class implement an interface? Explain your answer.

- Yes
- An abstract class can implement interfaces without even providing the implementation of all interface methods. It is the responsibility of the first concrete class that has that abstract class as an ancestor to implement all of the methods in the interface.

Discussion date: 14th September.
Facilitator: Dr. Janaka Alawathugoda