CO225: Software Construction
Tutorial 1

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

1. What are the OOP principles?

    - Inheritance
    - Polymorphism
    - Abstraction
    - Encapsulation

2. What is a class? What is the difference between a class and an object?

    - Class
        - A class is a blueprint to create different objects of same type. The class determines how an object will behave and what the object will contain.
    - Difference
        - A class is a template for objects. A class defines object properties including a valid range of values, and a default value. A class also describes the object behaviour.
        - But an object is an instance of a class. Object has a state in which all of its properties have values that you either explicitly define or that are defined by default settings.

3. Explain the difference between final and finally keyword.

    - final keyword is used to apply restrictions on a class, method or variable. Methods with final keyword are executed upon their call.
    - But finally keyword is used for a block which contains an important code that will be executed whether or not an exception is handled. This block is always associated with a try catch block. This block executes just after the execution of the try catch block.

4. What does the static keyword mean in Java?

    - When static keyword is applied to a member, which is a variable, method, block or nested class, that member can be accessed without requiring an instantiation of the class to which it belongs.

5. Explain the difference between interface and abstract class. Give two examples of where you would use an abstract class and interface.
    - Interface can have only abstract methods. But an abstract class can have abstract and non-abstract methods.
    - Interfaces has only static and final variables. But abstract class can have static, non-static, final and non-final variables.
    - To declare an interface ‘interface’ keyword is used. And to declare an abstract class ‘abstract’ keyword is used.

- An interface cannot provide the implementation of an abstract class but an abstract class can provide the implementation of an interface.
- Interface supports multiple inheritance. But abstract class doesn't support multiple inheritance.
- An interface should be implemented using 'implements' keyword. And an abstract class should be extended using 'extends' keyword.
- An interface can extend another Java interface only. But an abstract class can extend another Java class and implement multiple Java interfaces.
- Members of a Java interface are public by default. But a Java abstract class can have class members such as private, protected, etc.
- Abstract class example

```
abstract class Car {
        public void applyBrakes() {
          System.out.println("Apply brakes method is implemented");
        }
        public abstract void changeGears();
}
 // Now, any Car that wants to be instantiated must implement the changeGears () method.

class CarModel1 extends Car {
        public void changeGears() {
          System.out.println("Implement changeGears() method for Car model 1 Car");
        }
}
class CarModel2 extends Car {
        public void changeGears() {
          System.out.println("Implement changeGears() method for Car model 2 Car");
        }
}
class Test {
      public static void main(String[] args) {
            Car obj1 = new CarModel1();
            Car obj2 = new CarModel2();
            obj1.changeGears();
            obj2.changeGears();
      }
}
```

- Interface example

```
interface Bank{
        float rateOfInterest();
}
// Now, any bank that wants to be instantiated must implement the rateOfInterest () method.


class SBI implements Bank{
        public float rateOfInterest() {return 9.15f;}
}
class PNB implements Bank{
        public float rateOfInterest() {return 9.7f;}
}
class Test{
        public static void main(String[] args) {
                Bank obj1 = new SBI();
                System.out.println("Rate Of Interest: "+obj1.rateOfInterest());
                Bank obj2 = new PNB();
                System.out.println("Rate Of Interest: "+obj2.rateOfInterest());
        }
}
```

6. What are the two types in Polymorphism? (was not discussed in lectures. Search)

   - Compile time polymorphism
   - Runtime polymorphism

7. What is multi threading? Why is the use of it?

   - Multi threading
     - A process of executing two or more threads simultaneously.
   - Use
     - To provide simultaneous execution of two or more parts of a program to maximum utilize the CPU time.

8. What are the two types in thread creation? Why there are two types there?

   - Two types
     - By extending Thread class
     - By implementing Runnable interface.
   - Why there are 2 types
     - If we choose to implement Runnable interface then we could extend another class. Therefore implementing Runnable interface overcomes the limitation of inheriting from another class. But if we extends Thread class we cannot extend another class.

So depending on your design requirement we could use either of the methods.

9. Is multiple inheritance allowed in Java? What are the difficulties in multiple inheritance?

- No
- Difficulties
  - Leads to diamond problem of multiple inheritance.
    - Diamond problem
      - Let there be two classes B and C inheriting from A. Assume that B and C are overriding an inherited method and they provide their own implementation. Now D inherits from both B and C doing multiple inheritance. Therefore on calling the method, the compiler cannot determine which class method to be called and even on calling which class method gets the priority.
  - Complicates the design and it creates problem during various operations like casting, constructor chaining etc.
  - Rarely needed as there are very few scenarios on which we actually need multiple inheritance.

10. What is overloading and what are the advantages of overloading?

- Overloading is the ability to define more than one method with the same name in a class. The compiler distinguish between the methods using their method signatures.
- Advantages
  - Increases cleanliness of the code.
  - Increases the readability of the program.
  - Gives programmers the flexibility to call a similar method for different types of data.
  - Overloading is also used on constructors to create new objects given different amounts of data.
  - Methods can have different return types.

11. How many constructors can be there for a single class?

- 65535 constructors

12. How to call a method in a base class without creating an instance of that class?

- Declare the method in the base class as static using the static keyword.
  - Call the static method using the name of the base class as BaseClassName.StaticMethodName(arguments);

13. Given 3 classes, A, B, C and methodA() and methodB() in classes A and B respectively, how can you make sure that these 2 methods are available for use in class C?

- By creating instances of the classes A and B in class C and call the methods using the created instances. The access modifiers of the classes A and B should be public and classes A, B, C can be in different packages. Or the access modifiers of the classes A and B should be default and classes A, B, C must be in same package.

14. Explain the advantages of encapsulation.

- Encapsulated code is more flexible as we can make the variables of the class as read only or write only depending on our requirement.
    - If we need to make the variables read only then we don't define the setter methods.
    - And if we need to make the variables write only then we don't define the getter methods.
- Encapsulated code improves the re-usability and easy to change it with new requirements.
- The user will have no idea about the inner implementation of the class. It will not be visible to the user that how the class is storing values in the variables. This ensures security of the encapsulated code.
- Useful in hiding the data of a class, since outside classes will not be able to access the data through the private data members.
- You can change one part of the code without affecting other parts.
- Provides the control over the data.
- Easy to test the encapsulated class.

15. What is the difference between public class and public final class?

- A public class can be extended. But a public final class cannot be extended.

Discussion date: 25th August 2017
Facilitator: Dr. Janaka Alawathugoda