**07 - Abstract vs. Interface Questions | Ibad H.**

1. **What is an abstract class?**
   1. An abstract class is a class that cannot be instantiated as an object. It is used as a super class to create a level of abstraction for a series of objects that a program might contain. For example, the abstract class employee might have fields and methods that describe a general employee, but, in this case, it cannot exist as an instance in a program as it is simply a set of base details that fit different employee subclasses. A subclass of this abstract class could be project manager, which would be a specific instance of the employee abstract which could be instantiated in a program and given specific attributes. Overall, abstract classes are generalized classes used to organize a larger set of classes that share common factors.
      1. [new java\_abstraction.pdf](https://drive.google.com/file/d/1gVFVGBQnOTGhaovYmH1ByR-fsKcuoEjL/view)
2. **What is an abstract method?**
   1. An abstract method is a general method that is defined, within an interface, with a return type and any parameters (if needed), but doesn’t contain any implementation therefore being unable to perform anything. It is essentially the base skeleton for a method. The implementation for the method is usually specified in the child classes so each child class of the same interface can utilize the method of the same name differently.
      1. [new java\_abstraction.pdf](https://drive.google.com/file/d/1gVFVGBQnOTGhaovYmH1ByR-fsKcuoEjL/view)
3. **What is an interface?**
   1. An interface is a collection of abstract methods used to indicate a class’ behaviors. Rather than abstract classes which specify attributes and behaviors of an object, interfaces simply create behaviors that classes implement. The methods in an interface must be defined by each class therefore meaning that the same method from an interface can do very different things depending on the class that's implementing it. Essentially, interfaces are used to define a general behavior set for the several classes.
      1. [new java\_interfaces.pdf](https://drive.google.com/file/d/1LN-mTTefFUl56ryfA-5PaCy3ADru8KCb/view)
4. **What is the difference between an interface and an abstract class?**
   1. An interface is used to create a general behavior set for different objects, abstract classes are used to create a base object class that is instantiated through specific child classes that provide practical implementation of that abstract. While an abstract class is an abstraction of an object, an interface is a set of practices that different objects can follow. The key differences being that interfaces cannot contain attributes as they do not define an object.
      1. [new java\_interfaces.pdf](https://drive.google.com/file/d/1LN-mTTefFUl56ryfA-5PaCy3ADru8KCb/view)
5. **When should you use an interface vs. an abstract class?**
   1. Interfaces are best when you need multiple different classes (of very different types) to share some common behaviors, whereas abstract classes are most useful when you want to create a plethora of objects with common attributes and behaviors.
      1. [new java\_interfaces.pdf](https://drive.google.com/file/d/1LN-mTTefFUl56ryfA-5PaCy3ADru8KCb/view)