**134. What is an Abstract Class.**

1. There are two types of classes:

a. Abstract Class: This class is defined using “abstract”.

b. Concrete Class: Normal Class.

2. We cannot create the object of Abstract Class.

3. We can create the reference of the Abstract Class.

4. **Abstract Method:** A method is not having any body. Undefine method is known as Abstract Method.

5. A Class having at least on abstract method is wholly considered as abstract.

6. A abstract class may have zero or more abstract method.

7. A class which overrides every abstract method of the base class then that class becomes the concrete class otherwise it will become the abstract class even if one abstract method is not overridden.

8. An abstract method is only declared in the abstract class.

9. An abstract class are meant to overridden.

**10. Abstract classes are useful for defining and imposing standards.**

**135. Practicing Abstract Classes.**

**136. Example #1.**

1. Q1. Hospital

2. Abstract class is a body which approve the standards.

3. Sub classes of Abstract are meant to follow the standards.

**137. Example #2.**

1. Q1. KFC

2. Abstract classes are like blueprint to be followed while building the concrete classes.

3. We can create the reference of the abstract class(It like front board of a shop which belongs to someone having different brand name).

**138. Student Challenge: Class Shape.**

1. Abstract Class: Shape. (There are no specific properties of shapes but have common methods.)

2. Derived Classes: Circle, Rectangle.

3. circle:

properties: radius.

methods: area, perimeter.

4. rectangle:

properties: length, breadth.

methods: area, perimeter.

**139. Dos and Don’ts of Abstract Class and Methods.**

**a. I can have reference of it.**

**b. An abstract method should be defined with an “abstract” keyword.**

**c. Abstract method should be declared in abstract method.**

**d. Abstract classes cannot be final(It means classes cannot be extended).**

**e. It cannot create the object of it.**

**f. An abstract method cannot be final.**

**g. An abstract class cannot be static.**

**h. An abstract method cannot be static.**

**i. Sub class must override all the abstract methods inherited from the abstract class.**

**140.**