Abstraction:

Abstraction means Show the functionality and Hide the complexity.

{{ Aam Kha Ped Mat Geen. Kaam Kisi Aur Ka Karna Kisi Aur Ko Hai. }}

#. Definition:

Abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does.

In case of Abstraction two roles are important.

1. Developer/Technology that executes the work.
2. Programmer, who provides the work.

#. There are two ways to achieve Abstraction.

By **Abstract** class and by **Interface** class

Abstract Class

#. In case of Abstract class, there are few required rules that we have to follow to successfully achieve abstraction.

R-1: Abstraction can only be used via Inheritance (IS-A relationship), we cannot use it through **Association**.

R-2: A Child class has to **Register** itself with Parent class by passing Reference of a child class to a parent class.

R-3: For registration, a parent class or developer class has to provide a Method in which child is going to pass the Reference as argument.

R-4: Abstract class cannot be instantiated that means we cannot create object of an abstract class.

R-5: A Child has to override all the abstract method of an abstract class, other vise the child class has to make itself as an Abstract.

IQ: Can other classes can have abstract method?

Ans: No, Only Abstract class can have abstract method.

IQ: Is it mandatory to have at least one method is an abstract class?

Ans: It is not mandatory to have at least one abstract method in an abstract class.

IQ: Can we keep a constructor in abstract class?

Ans: YES we can.

IQ: Can we make an abstract method as a private?

Ans: No, because if we made private abstract method then we won't be able to override that private methods from outside the abstract class.

IQ: Can we make method in abstract class as private?

Ans: Yes From JDK9, we can have private method in abstract class static and non-static.

Static methods can use/call in main () method. Non-Static methods can call in constructor. We can also call those methods from child class object.

IQ: Can we make an abstract method as a static?

Ans: NO, we can't make abstract method as static because to achieve Abstraction Method Overriding is required and static functions do not override, they do function Hiding.

IQ: Can we make static method in abstract class.

Ans: Yes. We can make static method in abstract class. We can also have main method in abstract class that means abstract class can be executable.

IQ: What is called abstract method?

Ans: A method without body is abstract method.

Mine:

#. To achieve Abstraction, it requires to implement Method Overriding, UpCasting, and Dynamic Binding.

#. Method Overriding will be done by Programmer.

#. Demo Program of Abstraction using Abstract Class

//Technology Program

// To create abstract class, we must need to add "**abstract**" keyword before class declaration syntax.

```
abstract class AssignmentMaker{

        //no class can have without constructor.

        AssignmentMaker(){

                System.out.println("Parent Constructor");

        }

        // It is not require to have abstract method, abstract method does
not have body

        abstract void makeAssignment();

        //registration function. Providing registration method to
programmer to receiving the         child reference id

        public void CyberCafe(AssignmentMaker am){

                // Calling the overriden method that is in child class.

                am.makeAssignment();

        }

        //static methods are allowed

        static void show(){

                System.out.println("Static Show From Parent");

        }

        // From JDK 9, we can have private method in abstra class static and
non-static

                private static void methodPrivate(){

                        System.out.println("I am Private Method of Abstract Class");

                }

        // "Abstrac static" method are not allowed

                /*      abstract static void display();     */

        // We can make parent class excutable becasue static methods are
allowed
```

```
        public static void main(String...a){
                System.out.println("Executing The Abstrac Parent Class Direct");
                // also can call static method
                show();
        }
}
```

// Programmer Program

```
class Student extends AssignmentMaker{
        String assignment = "Java";
        //overriding the method defined in technlogy's abstact class
        void makeAssignment(){
                // hundred lines of code.
                System.out.println(" Code Sent");
        }
        public static void main(String...a){
```

// Creating child class obect

```
                Student s = new Student();
```

// calling register method of abstract class. Can call parent method from child class object beacuse extended the abstract class

```
                s.CyberCafe(s);
        }
}
```

IQ: Why can’t we create the object of an abstract class?

Ans: Because these classes are incomplete, they have abstract methods that have no body so if java allows you to create object of this class then if someone calls the abstract method using that object then what would happen? There would be no actual implementation of the method to invoke.

Also because an object is concrete. An abstract class is like a template, so you have to extend it and build on it before you can use it.

Unresolved compilation problem: Cannot instantiate the type AbstractDemo

IQ: Can we make abstract method as final? Why?

Ans: No, we can't make abstract method as final because we define abstract method for future improvement and once we define it as final, we won't be able to override that method.

IQ: Can we instantiate/make object an abstract class which does not have even a single abstract method?

Ans: No, once we declared a class as abstract, we can't instantiate.

IQ: We can't instantiate the abstract class then why constructors are allowed in abstract class.

Ans: we can't instantiate the abstract class but their sub classes can instantiate, from sub class we can call the abstract class constructor.

IQ: Can a class contain an abstract class as a member?

Ans: Yes, a class can have an abstract class as its member. Even abstract class can be nested that means an abstract class can have another class as its member.

IQ: Can Abstract classes can be nested. ?

Ans: Yes, Abstract classes can be nested i.e. an abstract class can have another abstract class as its member.

IQ: Can we declare local inner class as abstract?

Yes. Local inner class can be abstract.

```
class ClassOne{
	// Nested abstract class
	abstract class ClassTwo{
		// Nested Abstract class
		abstract class ClassThree{
		}
	}
	public void display(){
		// local inner class as abstract
		abstract class ClassTwo{
			abstract class ClassThree{
				// Nested Abstract class
			}
		}
	}
	public static void main(String...a){
	}
}
```

IQ: Can abstract method declaration include throws clause?

Ans: Yes. Abstract methods can be declared with throws clause.