**what is abstraction?**

The meaning of the word “Abstraction”, in general words, is the process of working with ideas rather than their implementation.

For example, consider the example of an email, the user does not know about the complex details such as what happens just after sending an email, which protocol is used by the server to send the message.

Therefore, we just need to mention the address of the receiver, type the content and click the send button.

This is basically called Abstraction in which the complex details are being hidden from the users.

Similarly, in Object-oriented programming, abstraction is a process of providing functionality to the users by hiding its implementation details from them.

In other words, the user will have just the knowledge of what an entity is doing instead of its internal working.

**Is Abstract class is related to Abstraction? find it?**

Abstract class and Abstraction both are different concept.

Abstract class provides the flexibility to have certain concrete methods and some other methods that the derived classes should implement.

Abstract class have normal methods and abstract methods.

Main purpose of abstract class is two things 1. Reusability 2. Enforcement

Abstraction-it is one of the pillar of oops concept. Abstraction is a process of hiding the implementation details and showing only functionality to the user.

another way, it shows only essential things to user and hides the internal details.

**can interface have normal variable ?**

We keep those variables within the interface which are fixed or else we can say we keep those variables within the interface class which the developers should not forget any given cost. Those variable are called interface variables. Which is a pure abstract class.

**Can interfaces have normal variables ?**

Interface has no state, all variables declared inside interface are implicitly public, static and final.

**Can interfaces have methods with definitions ? if so how ? or what ?**

Like a class, an interface can have methods and variables, but the methods declared in an interface are by default abstract (only method signature, no body).

Interfaces specify what a class must do and not how. It is the blueprint of the class.

**How to implement multiple inheritance using interfaces**

An interface contains variables and methods like a class but the methods in an interface are abstract by default unlike a class.

Multiple inheritance by interface occurs if a class implements multiple interfaces or also if an interface itself extends multiple interfaces.

interface AnimalEat {

void eat();

}

interface AnimalTravel {

void travel();

}

class Animal implements AnimalEat, AnimalTravel {

public void eat() {

System.out.println("Animal is eating");

}

public void travel() {

System.out.println("Animal is travelling");

}

}

public class Demo {

public static void main(String args[]) {

Animal a = new Animal();

a.eat();

a.travel();

}

}

Output

Animal is eating

Animal is travelling

Now let us understand the above program.

The interface AnimalEat and AnimalTravel have one abstract method each i.e. eat() and travel().

The class Animal implements the interfaces AnimalEat and AnimalTravel. A code snippet which demonstrates this is as follows:

interface AnimalEat {

void eat();

}

interface AnimalTravel {

void travel();

}

class Animal implements AnimalEat, AnimalTravel {

public void eat() {

System.out.println("Animal is eating");

}

public void travel() {

System.out.println("Animal is travelling");

}

}

In the method main() in class Demo, an object a of class Animal is created.

Then the methods eat() and travel() are called. A code snippet which demonstrates this is as follows:

public class Demo {

public static void main(String args[]) {

Animal a = new Animal();

a.eat();

a.travel();

}

}