**Question 1: what is abstraction in java?**

Abstraction is the process of hiding the implemented code from the user . only the highlighted set of service provided to the user .



|  |
| --- |
| 1. Add 2. Subtraction 3. Multiplication 4. Division |



User



In the process of abstraction, the user doesn’t know which code is running in the backend. but the output what the person wants (for example, I have taken the example of basic calculator) can access.

**Question 2: Is abstract class related to abstraction ?**

Abstract classes/methods are created so that it can be **implemented in its subclasses** because the abstract class does not know what to implement in the method but it knows that the method will exist in its subclass. So, while we creating subclass we need to **override the abstract method** to provide its implementation.

Where as abstraction is the process of hiding the implemented code from the user. Abstraction is similar to the concept of a black box. Input goes in, black box does something, output comes out. It doesn't matter what happens in the black box, all we have to know is that it works. That’s all .

**Question 3 : 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.

Question 4 : can interfaces have methods with definations ?

Interface is used to declare a set of constants that can be used in different classes. Since the interface do not contain any methods, these constant values will be available to any class that implements interfaces. The value can be used in any methods as a part of the variable declaration.

Validation code :

public interface area1 {

float ***z*** = 3.14f;

int ***m*** =10 ;

}

class area2 implements area1

{

int c = ***m***;

float x = ***z***\*c;

void size( int size) {

if(size > x) {

System.***out***.println(" come in ");

}

else{

System.***out***.println(" gate out " );

}

}

}

public class main {

public static void main(String[] args) {

area2 area = new area2();

area.size(4);

}

}

**Question 5 : How to implement multiple inheritance using interface ??**

package multiinheritance;

public class student {

int rollNumber;

void getNumber(int n ) {

rollNumber=n;

}

void putNumber() {

System.***out***.println( " roll number : " + rollNumber);

}

}

class test extends student{

float part1, part2;

void getmarks(float m1, float m2) {

part1 = m1;

part2 = m2 ;

}

void putMarks() {

System.***out***.println( "marks obtained " );

System.***out***.println( " marks obtained in part 1" +part1);

System.***out***.println( " marks obtained in part 1" +part2);

}

}

package multiinheritance;

public interface sports {

float ***sportwt*** = 6.0f;

}

class Result extends test implements sports{

float total;

void putwt() {

System.***out***.println(" the sport wt is "+***sportwt*** );

}

void display()

{

total = part1 + part2 + ***sportwt***;

System.***out***.println(" the total score is " + total );

}

}

package multiinheritance;

public class main {

public static void main(String[] args) {

Result res = new Result();

res.getNumber(1110);

res.getmarks(87f,90f);

res.display();

}

}