* **QUESTIONS ON ABSTRACTION, ENCAPSULATION & INHERITANCE**

**Q.1 What is ‘Astraction’ & ‘Abstraction Class’ in Java?**

**Abstraction** is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery. Abstraction lets you focus on what the [object](https://www.javatpoint.com/object-and-class-in-java) does instead of how it does it.

**Abstraction Class in Java** is a class that is declared with an abstract keyword.

**Q.2** **What is the difference between abstract class and concrete class?**

There are mainly two differences between an abstract class and concrete class. They are:

a) We cannot create an object of abstract class. Only objects of its non-abstract (or concrete) sub classes can be created.

b) It can have zero or more abstract methods that are not allowed in a non-abstract class (concrete class).

**Q.3** **Can we define an abstract method inside non-abstract class (concrete class)?**

**No**, we cannot define an abstract method in non-abstract class.

For example:

class Test {

abstract void show();

}

The above code will generate a compile – time error.

**Q.4** **Which among the following code have errors?**

**a)** abstract class A {

void m1();

}

**b)** public class A {

abstract void m1();

}

**c)** abstract public class A {

abstract void m1();

}

**d)** abstract public class A {

abstract void m1();

A(){}

Void m1(){}

}

**f)** public abstract class A {

abstract int x = 100;

abstract void m1();

}

**h)** public abstract class A {

abstract void m1();

A(){}

Static void m2() {

System.out.println(“Hello Java”);

}

public class B extends A {

void m1(){

A.m2();

}

}

**i)** public abstract classs A {

abstract void m1();

private A(){}

}

Public class B extends A {}

**Ans: a, b, f, g, i.**

**Q.5** **What is the advantage of Abstract class in Java?**

The main advantages of using abstract class are as follows:

* Abstract class makes programming better and more flexible by giving the scope of implementing abstract methods.
* Programmer can implement abstract method to perform different tasks depending on the need.
* We can easily manage code.

Ex.1

Prog.:- abstract class Bike{

abstract void run();

}

class Honda4 extends Bike{

void run(){System.out.println("running safely");

}

public static void main(String args[]){

Bike obj = new Honda4();

obj.run();

}

}

Ex.2

abstract class Shape{

abstract void draw();

}

//In real scenario, implementation is provided by others i.e. unknown by end user

class Rectangle extends Shape{

void draw(){System.out.println("drawing rectangle");

}

}

class Circle1 extends Shape{

void draw(){System.out.println("drawing circle");

}

}

//In real scenario, method is called by programmer or user

class TestAbstraction1{

public static void main(String args[]){

Shape s=new Circle1();//In a real scenario, object is provided through method, e.g., getShape() method

s.draw();

}

}