

## MCQ Exam Result

### Result Summary

Field	Value
Test ID	41212
Total Questions	20
Marks Obtained	19
Attempted	20
Non-Attempted	0
Percentage	95.00%
Grade	Outstanding

### Question Details

Q.No	Question	Your Answer	Correct Answer	Result	Status
1	Can we change the return type of an overriding method be a subclass of the original method's return type?	Yes, covariant return type	Yes, covariant return type	Correct	Attempted
2	Find the output:  <pre> abstract class X {     abstract void a();     abstract void b(); } class Y extends X {     void a() { System.out.print("A"); }     void b() { System.out.print("B"); } } public class Test {     public static void main(String[] args) {         X obj = new Y();         obj.a();         obj.b();     } } </pre>	AB	AB	Correct	Attempted
3	Find the Output question:?  <pre> class A {     protected void display() {         System.out.println("A display");     } } class B extends A {     public void display() {         System.out.println("B display");     } } public class Test {     public static void main(String[] args) { </pre>	B display	B display	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre> A obj = new B(); obj.display(); } } </pre>				
4	Which keyword is used to declare an interface in Java?	interface	interface	Correct	Attempted
5	Can an abstract class be final?	No	No	Correct	Attempted
6	Can a subclass override a static method?	No	No	Correct	Attempted
7	How does a class implement an interface?	Using implements	Using implements	Correct	Attempted
8	Choose the answer?:  <pre> class Parent {     final void show() {         System.out.println("Parent"); }     } class Child extends Parent {     void show() { System.out.println("Child");     } } </pre> Note:- just analyze the syntax:	Compilation Error	Compilation Error	Correct	Attempted
9	Which of the following is true about method overriding?	Overridden method must have the same signature	Overridden method must have the same signature	Correct	Attempted
10	Can an abstract method be static?	Yes	No	Incorrect	Attempted
11	Which of the following is true about methods in a Java interface?	Methods are abstract by default	Methods are abstract by default	Correct	Attempted
12	Can a class implement multiple interfaces in Java?	Yes	Yes	Correct	Attempted
13	Can an abstract class extend another abstract class?	Yes	Yes	Correct	Attempted
14	What is the purpose of declaring a method abstract?	To enforce subclass implementation	To enforce subclass implementation	Correct	Attempted
15	Which of the following is not allowed in an abstract class?	Abstract method with private access	Abstract method with private access	Correct	Attempted
16	Find the Output question?:  <pre> class Parent {     static void greet() {         System.out.println("Parent static"); }     void say() { System.out.println("Parent instance"); }     } class Child extends Parent {     static void greet() {         System.out.println("Child static"); }     void say() { System.out.println("Child instance"); }     } public class Test {     public static void main(String[] args) {         Parent p = new Child();         p.greet();         p.say();     } } </pre>	Parent static Child instance	Parent static Child instance	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre>     }   } </pre>				
17	<p>find the OutPut of below code?</p> <pre> class Parent {     void show() {         System.out.println("Parent Show");     } } class Child extends Parent {     void show() {         System.out.println("Child Show");     } } public class Test {     public static void main(String[] args) {         Parent p = new Child();         p.show();     } } </pre>	Child Show	Child Show	Correct	Attempted
18	<p>Can an interface contain a constructor in Java?</p>	No	No	Correct	Attempted
19	<p>What is printed?</p> <pre> class Parent {     static void greet() {         System.out.println("Parent"); } } class Child extends Parent {     static void greet() {         System.out.println("Child"); } }  class Main{     public static void main(String[] args){         Parent p = new Child();         p.greet();     } } </pre>	Parent	Parent	Correct	Attempted
20	<p>Find the output:</p> <pre> abstract class A {     abstract void msg(); } class B extends A {     void msg() {         System.out.println("Hello");     } } public class Test {     public static void main(String[] args) {         A obj = new B();         obj.msg();     } } </pre>	Hello	Hello	Correct	Attempted