SIMULADORES

WORKING WITH CONSTRUCTORS, METHODS AND ENCAPSULATION

1.

Test Overview

Name	Taken on - 01 ago., '24 11:58 a. m.	**Status**	Passed 79%
Correct Answers	15	**Total Questions**	19
Time Taken	01:09:50	**Total Time**	00:40:32
Start Time	01 ago. 24 11:58	**Finish/Pause Time**	01 ago. 24 13:36

Test Details | Performance Report

S ...	Marked	Atte...	Result	Exam Objective	Difficulty Le...	Problem Statement	Note
1		✓	✓	06 - Constructors	Very Easy	Given a class named Test, which of these wo...	
2		✓	✓	06 - Working with ...	Easy	System.out.println("In integer	
3		✓	✓	01 - Java Basics - O...	Easy	When a class, whose members should be acc...	
4		✓	✓	06 - Working with ...	Very Easy	}	
5		✓	✓	06 - Working with ...	Very Easy	public class OverloadTest{	
6		✓	✓	01 - Java Basics - O...	Tough	private void computeAverage(){	
7		✓	✓	06 - Working with ...	Very Easy	}	
8		✓	✓	06 - Working with ...	Easy	return time*speed ;}	
9		✓	✓	06 - Working with ...	Very Easy	Which of the following methods does not ret...	
10		✓	✓	06 - Working with ...	Easy	processStacks (s1,s2);	
11		✓	✗	06 - Working with ...	Easy	ref = new TestClass();	
12		✓	✓	01 - Java Basics - O...	Very Easy	What is meant by "encapsulation" ?	
13		✓	✗	06 - Working with ...	Very Tough	public RETURNTYPE methodX(byte by){	
14		✓	✗	01 - Java Basics - O...	Very Tough	Identify correct option(s)	
15		✓	✓	06 - Working with ...	Very Easy	s1 = sM1("c");	
16		✓	✓	06 - Working with ...	Easy	public static double getAngle();	
17		✓	✓	06 - Working with ...	Very Easy	a = 20;	
18		✓	✓	06 - Working with ...	Tough	a.link = b;	
19		✓	✗	06 - Working with ...	Real Brainer	static DatabaseWrapper getDatabase()	

2.

WORKING WITH INHERITANCE

1.

Test Overview

Name	Taken on - 01 ago., '24 04:09 p. m.	**Status**	Passed 65%
Correct Answers	13	**Total Questions**	20
Time Taken	00:44:59	**Total Time**	00:42:40
Start Time	01 ago. 24 16:09	**Finish/Pause Time**	01 ago. 24 16:58

Test Details | Performance Report

S ...	Marked	Atte...	Result	Exam Objective	Difficulty Le...	Problem Statement	Note
1		✓	✓	07 - Working with I...	Easy	a method overriding the given method:	
2		✓	✗	07 - Working with I...	Easy	A o1 = new C( );	
3		✓	✗	07 - Working with I...	Tough		
4		✓	✓	07 - Working with I...	Easy		
5		✓	✗	07 - Working with I...	Tough	A(int i) { this.i = i; }	
6		✓	✓	07 - Working with I...	Very Easy	Automobile a = new Automobile();	
7		✓	✗	07 - Working with I...	Very Easy		
8		✓	✗	07 - Working with I...	Very Easy	B b = new B();	
9		✓	✓	07 - Working with I...	Easy	class Base2 extends Base{	
10		✓	✗	07 - Working with I...	Tough	public class TestClass {}	
11		✓	✓	07 - Working with I...	Very Tough	public void methodA(){	
12		✓	✓	07 - Working with I...	Easy	}	
13		✓	✓	07 - Working with I...	Very Tough		
14		✓	✓	07 - Working with I...	Easy	public byte getValue(){ return 2; } //2	
15		✓	✓	07 - Working with I...	Very Easy	}	
16		✓	✓	07 - Working with I...	Very Tough	+-- enthu.trans.DataFlooding	
17		✓	✓	07 - Working with I...	Easy	class B extends A{	
18		✓	✗	07 - Working with I...	Tough	}	
19		✓	✓	07 - Working with I...	Easy	}	
20		✓	✓	07 - Working with I...	Very Easy	List students = new ArrayList();	

2.

HANDLING EXCEPTIONS