MCQ Exam Result

Result Summary

Field	Value
Test ID	39054
Total Questions	30
Marks Obtained	30
Attempted	30
Non-Attempted	0
Percentage	100.00%
Grade	Outstanding

Question Details

Q.No	Question	Your Answer	Correct Answer	Result	Status
1	If a class has only private constructors, it means:	The class cannot be instantiated from outside	The class cannot be instantiated from outside	Correct	Attempted
2	What is the output? class Student { int marks; Student(int marks) { this.marks = this.marks + 10; } void show() { System.out.println("Marks: " + marks); } public static void main(String[] args) { Student s = new Student(50); s.show(); } }	Compilation error(inappropriate use of this)	Compilation error(inappropriate use of this)	Correct	Attempted
3	What will be the output? class Account { int balance = 100; void update() { int balance = 50; this.balance += balance; } public static void main(String[] args) {	150	150	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	Account a = new Account(); a.update(); System.out.println(a.balance); } }				
4	Predict the output: class Box { int length; Box(int length) { this.length=9; this.length = length; length = 5; } public static void main(String[] args) { System.out.println(new Box().length); } }	Compilation error	Compilation error	Correct	Attempted
5	class Demo { void display() { System.out.println(this); } public static void main(String[] args) { Demo d = new Demo(); d.display(); } }	Prints: Demo@<hashcode>	Prints: Demo@<hashcode>	Correct	Attempted
6	Which of the following statements about Encapsulation in Java are TRUE? i. Encapsulation is the process of hiding data implementation details using access modifiers like private. ii. Encapsulation can be achieved using private variables and public setter/getter methods. iii. Encapsulation and Abstraction are exactly the same concept. iv. Encapsulation allows direct access to instance variables from outside the class.	i and ii	i and ii	Correct	Attempted
7	Find the incorrect usage of this: class Employee { private int id; private String name; public static void setData(int id, String name) { this.id = id; this.name = name; } }	b and c both	b and c both	Correct	Attempted
8	What is the output? class Student { int marks;	Compilation error(inappropriate use of this)	Compilation error(inappropriate use of this)	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	Student(int marks) { this.marks = this.marks + 10; } void show() { System.out.println("Marks: " + marks); } public static void main(String[] args) { Student s = new Student(50); s.show(); } }				
9	class Employee { private int salary; public void setSalary(int salary) { salary = this.salary; } public int getSalary() { return salary; } } public class TestYourAbility { public static void main(String[] args) { Employee e = new Employee(); e.setSalary(5000); int option = 2; switch (option) { case 1: System.out.println("Salary: " + e.getSalary()); break; case 2: System.out.println("Salary: " + e.getSalary()); break; default: System.out.println("No data"); } } } What is the output?	Salary: 0	Salary: 0	Correct	Attempted
10	What does this() represent inside a constructor?	represent instance of that class	represent instance of that class	Correct	Attempted
11	What will be the output of the following code? class Test { int x; Test(int x) { this.x = x; x=20 } public static void main(String[] args) { int y=18; this .x=y; System.out.println(new Test().x); } }	Compile-time error	Compile-time error	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
12	Identify the error in this program: class Demo { int value; static void printValue() { System.out.println(this.value); } }	this.value cannot be used in static method	this.value cannot be used in static method	Correct	Attempted
13	What will happen here? What will happen here? class Student { int marks; void assignMarks(int marks) { this.marks = marks; } void copyMarks(Student s) { this = s; } }	Compile-time error: cannot assign a value to this	Compile-time error: cannot assign a value to this	Correct	Attempted
14	Why do we use getter and setter methods?	To provide controlled access to private data members	To provide controlled access to private data members	Correct	Attempted
15	What will be the output? class Account { int balance = 100; void update() { int balance = 50; this.balance += balance; } public static void main(String[] args) { Account a = new Account(); a.update(); System.out.println(a.balance); } }	150	150	Correct	Attempted
16	What does this() represent inside a constructor?	represent instance of that class	represent instance of that class	Correct	Attempted
17	What will be the output? ```java class Test { Test() { System.out.println("Default"); } Test(int a) { System.out.println("Parameterized"); } public static void main(String[] args) { new Test(5); } } ```	Parameterized	Parameterized	Correct	Attempted
18	What will be the output of the following code?	Compile-time error	Compile-time error	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	class Test { int x; Test(int x) { this.x = x; x=20 } public static void main(String[] args) { int y=18; this .x=y; System.out.println(new Test().x); } }				
19	Which of the following correctly defines a copy constructor-like behavior in Java? ```java class Test { int x; Test(int val) { x = val; } Test(Test t) { x = t.x; } } ```	It copies the value of data members from one object to another	It copies the value of data members from one object to another	Correct	Attempted
20	Find the incorrect usage of this: class Employee { private int id; private String name; public static void setData(int id, String name) { this.id = id; this.name = name; } }	b and c both	b and c both	Correct	Attempted
21	What will happen here? What will happen here? class Student { int marks; void assignMarks(int marks) { this.marks = marks; } void copyMarks(Student s) { this = s; } }	Compile-time error: cannot assign a value to this	Compile-time error: cannot assign a value to this	Correct	Attempted
22	Identify the error in this program: class Demo { int value; static void printValue() { System.out.println(this.value); } }	this.value cannot be used in static method	this.value cannot be used in static method	Correct	Attempted
23	Which of the following demonstrates encapsulation properly?	Using private variables with public getters and setters	Using private variables with public getters and setters	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
24	Predict the output: class Box { int length; Box(int length) { this.length=9; this.length = length; length = 5; } public static void main(String[] args) { System.out.println(new Box().length); } }	Compilation error	Compilation error	Correct	Attempted
25	class Demo { void display() { System.out.println(this); } public static void main(String[] args) { Demo d = new Demo(); d.display(); } }	Prints: Demo@<hashcode>	Prints: Demo@<hashcode>	Correct	Attempted
26	Identify the error: ```java class Demo { private int a; void class(int x) { a = x; } } ```	Invalid syntax - Compilation Error	Invalid syntax - Compilation Error	Correct	Attempted
27	Where can the this keyword not be used?	Inside static blocks	Inside static blocks	Correct	Attempted
28	class Account { private double balance; Account(double balance) { this.balance = this.balance; } public double getBalance() { return balance; } } publicclass TestYourAbility { public static void main(String[] args) { Account a = new Account(1000.0); System.out.println("Balance: " + a.getBalance()); } } What will be the output?	Balance: 0.0	Balance: 0.0	Correct	Attempted
29	Where can the this keyword not be used?	Inside static blocks	Inside static blocks	Correct	Attempted
30	class Demo { int x;	i, iii	i, iii	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	static int y; Demo() { this.x = 10; this.y = 20; } } Statements: i. this.x is valid. ii. compilation error iii. Static variables can be accessed without this. iv. Using this for static variable is discouraged and illegal.				