

MCQ Exam Result

Result Summary

Field	Value
Test ID	39039
Total Questions	30
Marks Obtained	13
Attempted	30
Non-Attempted	0
Percentage	43.33%
Grade	Pass

Question Details

Q.No	Question	Your Answer	Correct Answer	Result	Status
1	<p>what is the specific output of this program ?</p> <pre> class Student { int marks = 50; void calculate() { int total = marks + 25; int this = 100; System.out.println("Total Marks: " + total); } public static void main(String[] args) { Student s = new Student(); s.calculate(); } } </pre>	Compilation error: cannot find symbol this	Compilation error: not a statement	Incorrect	Attempted
2	<p>Which of the following statements about Encapsulation in Java are TRUE?</p> <p>i. Encapsulation is the process of hiding data implementation details using access modifiers like private.</p> <p>ii. Encapsulation can be achieved using private variables and public setter/getter methods.</p> <p>iii. Encapsulation and Abstraction are exactly the same concept.</p> <p>iv. Encapsulation allows direct access to instance variables from outside the class.</p>	i and ii	i and ii	Correct	Attempted
3	<p>What does the expression this refer to when used inside a constructor?</p>	Current class instance being created	Current class instance being created	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
4	<pre> class Pen { Pen(int price) {} } public class Test { public static void main(String[] args) { new Pen(); } } </pre>	Compile time error: no default constructor	Compile time error: no default constructor	Correct	Attempted
5	<pre> class Employee { private int id; private String name; Employee(int id, String name) { this.id = id; this.name = name; } Employee(Employee e) { this = e; } } </pre> <p>Which statements are true?</p> <p>i. Copy constructor is valid. ii. this = e; is illegal. iii. Copy constructor should use this.id = e.id; etc. iv. Assigning this in a constructor leads to compile-time error.</p>	ii and iii only	ii, iii, iv	Incorrect	Attempted
6	<pre> public class Employee { private String name; private int age; Employee(String name, int age) { this.name = name; this.age = age; } Employee(Employee e) { this = e; } public void setName(String name) { this.name = name; } public String getName() { return this.name; } public static void showDetails() { System.out.println(this.name + " : " + this.age); } public void display() { this = new Employee("Temp", 25); System.out.println("Employee: " + this.name); } public static void main(String[] args) { Employee e1 = new Employee("John", 30); Employee e2 = new Employee(e1); Employee.showDetails(); e1.display(); } } </pre> <p>What will happen when the above code is compiled?</p> <p>i) Compile-time error in the copy constructor because this cannot be</p>	(i), (ii), and (iii)	(i), (ii), and (iii)	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<p>assigned a new reference.</p> <p>ii) Compile-time error in showDetails() because this cannot be used inside a static context.</p> <p>iii) Compile-time error in display() because this cannot be reinitialized or assigned a new object.</p> <p>iv) The program compiles successfully</p>				
7	<pre>class Student { private int marks; Student(Student s) { this.marks = s.marks; } }</pre> <p>Statements:</p> <p>i. Copy constructor can access private fields of same class.</p> <p>ii. Copy constructor cannot access private fields of another class.</p> <p>iii. this keyword is required for field assignment.</p> <p>iv. Copy constructor automatically deep copies objects.</p>	i, ii, iii	i, iii	Incorrect	Attempted
8	<pre>class TestYourAbility { TestYourAbility() { System.out.println("Constructor Called"); TestYourAbility obj = new TestYourAbility(5); System.out.println("End of Constructor"); } TestYourAbility(int x) { System.out.println("Parameterized Constructor: " + x); } public static void main(String[] args) { new TestYourAbility(); System.out.println("Main Ends"); } }</pre>	<p>Constructor Called</p> <p>? Parameterized</p> <p>Constructor: 5 ?</p> <p>End of Constructor</p> <p>? Main Ends</p>	<p>Constructor Called</p> <p>? Parameterized</p> <p>Constructor: 5 ?</p> <p>End of Constructor</p> <p>? Main Ends</p>	Correct	Attempted
9	<pre>public class TestYourAbility { int value = 1; TestYourAbility() { System.out.println("Inside Constructor: value = " + value); TestYourAbility obj = new TestYourAbility(); System.out.println("Constructor Completed: value = " + value); } public static void main(String[] args) { System.out.println("Main Started"); TestYourAbility t = new TestYourAbility(); System.out.println("Main Ended"); } }</pre>	Runtime Error - StackOverflowError.	Runtime Error - StackOverflowError.	Correct	Attempted
10	<p>What will be the output?</p> <pre>class Student { int marks;</pre>	<p>Total marks after bonus: 80</p> <p>Student reset completed!</p>	N/A	Incorrect	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre> Student(int marks) { this.marks = marks; } void resetStudent() { this = new Student(100); System.out.println("Student reset completed!"); } void calculateAverage() { int bonus = 10; int total = marks + bonus; System.out.println("Total marks after bonus: " + total); } public static void main(String[] args) { Student s = new Student(70); s.calculateAverage(); s.resetStudent(); } </pre>				
11	<pre> class Employee { int id; Employee(int id) { this.id = id; } Employee(Employee e) { e.id = 200; } } public class Test { public static void main(String[] args) { Employee e1 = new Employee(100); Employee e2 = new Employee(e1); System.out.println(e1.id + " " + e2.id); } } </pre>	200 200	200 0	Incorrect	Attempted
12	<pre> class Demo { int x; void Demo() { x = 10; } } </pre> <p>Statements:</p> <p>i. This is a constructor. ii. Return type void makes it a regular method. iii. Default constructor is still generated by compiler. iv. Compiler does not treat void Demo() as constructor.</p>	ii, iii, iv	ii and iv	Incorrect	Attempted
13	<p>Consider the statements:</p> <p>i. Every class in Java has a constructor. ii. If no constructor is defined, the compiler provides a default constructor. iii. Constructors can have a return statement. iv. A constructor can be any name.</p>	i, ii, iii	i, ii	Incorrect	Attempted
14	<p>What is the output of this program ?</p> <pre> class Test1 { private int number; static void setNumber(int number) { this.number=number; } } </pre>	Compile-time error	Compile-time error	Correct	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre> public static void main(String[] args) { new Test1().setNumber(10); number=12; System.out.println(number); } } </pre>				
15	<p>Default Constructor Characteristics :</p> <p>i. The default constructor initializes all numeric fields to 0 and object references to null. ii. The default constructor can have parameters. iii. The default constructor is provided by the compiler if no constructor is written. iv. The default constructor can contain user-defined code.</p>	i, iii	i, iii	Correct	Attempted
16	<p>Rules for Constructor Declaration</p> <p>Which of the following is invalid for a constructor declaration in Java?</p> <p>i. A constructor can have a return type of void. ii. A constructor name must match the class name. iii. A constructor can be overloaded. iv. A constructor can be declared private.</p>	All are valid	i only	Incorrect	Attempted
17	<pre> class BankAccount { private int balance; void setBalance(int balance) { if (balance > 0) balance = balance; // suspicious line else System.out.println("Invalid amount"); } int getBalance() { return balance; } } public class TestYourAbility { public static void main(String[] args) { BankAccount b = new BankAccount(); b.setBalance(1000); if (b.getBalance() == 0) System.out.println("Balance not updated"); else if (b.getBalance() > 0) System.out.println("Balance Updated"); else System.out.println("Invalid"); System.out.println("Final Balance: " + b.getBalance()); } } </pre> <p>What will be the output of the above program?</p>	Balance Updated Final Balance: 1000	Balance not updated Final Balance: 0	Incorrect	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
18	<pre>class Demo { int x; static int y; Demo() { this.x = 10; this.y = 20; } }</pre> <p>Statements:</p> <p>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.</p>	a) i, iii	a) i, iii	Correct	Attempted
19	<p>What will be the output of the following Java program?</p> <pre>class access { public int x; static int y; void cal(int a, int b) { x += a; y += b; } } class Main { public static void main(String args[]) { access obj1 = new access(); access obj2 = new access(); obj1.x = 0; obj1.y = 0; obj1.cal(1, 2); obj2.x = 0; obj2.cal(2, 3); System.out.println(obj1.x + " " + obj2.y); } }</pre>	b) 2 3	d) 1 5	Incorrect	Attempted
20	<pre>class Student { private String name; private int marks; public void setName(String name) { this.name = name; } public void setMarks(int marks) { marks = marks; } public String getName() { return this.name; } public int getMarks() { return this.marks; } }</pre> <p>Which of the following statements are true?</p> <p>i. this.name = name; correctly assigns the value to instance variable. ii. marks = marks; correctly assigns the value to instance variable. iii. getName() returns the current object's name. iv. getMarks() returns the correct marks after calling setMarks(50).</p>	i, iii, and iv	i and iii only	Incorrect	Attempted
21	<pre>class Book { String title; Book(String t) { title = t; } Book(Book b) { title = b.title; } }</pre>	Python Python	Java Python	Incorrect	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre> public class Test { public static void main(String[] args) { Book b1 = new Book("Java"); Book b2 = new Book(b1); b2.title = "Python"; System.out.println(b1.title + " " + b2.title); } } </pre>				
22	<pre> class Account { private double balance; Account(double balance) { balance = balance; // suspicious line } 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()); } } </pre> <p>What will be the output?</p>	Balance: 1000.0	Balance: 0.0	Incorrect	Attempted
23	What is the primary role of a setter method?	b) To set or update the value of a private attribute	b) To set or update the value of a private attribute	Correct	Attempted
24	<p>Which of the following statements are true?</p> <pre> class Box { private int length; void setLength(int length) { length = length; } } </pre> <p>Statements:</p> <p>i. Setter correctly sets instance variable. ii. length = length; does not update instance variable. iii. Must use this.length = length; iv. Getter is not affected by this mistake.</p>	ii, iii	ii, iii, iv	Incorrect	Attempted
25	<pre> class Test{ private int a = 10; public int method(int a){ a += 1; System.out.println(++a); return a; } } public class A public static void main(String args[]){ Test t = new Test(); t.method(3); System.out.Println(t.a); } } </pre>	c) compile time error	c) compile time error	Correct	Attempted
26	class Box {	i, ii, iv	i, iii, iv	Incorrect	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre>int length; }</pre> <p>Statements:</p> <ul style="list-style-type: none"> i. Class has a default constructor. ii. The default constructor takes arguments. iii. If any constructor is defined, default constructor is not provided. iv. Default constructor initializes fields to default values. 				
27	Where can the this keyword be legally used in Java?	b) Inside instance methods and constructors	b) Inside instance methods and constructors	Correct	Attempted
28	<pre>class Student { private int marks; public void setMarks(int marks) { marks = marks; } public int getMarks() { return marks; } public static void main(String[] args) { Student s = new Student(); s.setMarks(90); System.out.println(s.marks); } }</pre>	0	0	Correct	Attempted
29	<pre>class Student { private int id; private String name; private Student(int id, String name) { this.id = id; this.name = name; } protected int getId() { return id; } protected void setId(int id) { this.id = id; } protected String getName() { return name; } protected void setName(String name) { this.name = name; } } public class TestMcq { public static void main(String[] args) { Student s1 = new Studnet(101,"Virat"); Student s2 = new Studnet(909,"Rohit"); System.out.println(s1.getId() +" "+s2.getName()); } }</pre>	101 Virat	Compile time error	Incorrect	Attempted
30	<pre>class Employee { private int salary; public void setSalary(int salary) { this.salary = salary; } public int getSalary() { return salary; } }</pre>	Salary: 5000	Compile-time error: salary has private access in Employee	Incorrect	Attempted

Q.No	Question	Your Answer	Correct Answer	Result	Status
	<pre> } } 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.salary); break; case 2: System.out.println("Salary: " + e.getSalary()); break; default: System.out.println("No data"); } } } </pre> <p>What is the output?</p>				