| Q type | Level | Category | Skill | SubSkill | Topic | Q text | Option/ Answer 1 | Option/ Answer 2 | Option/ Answer 3 | Option/ Answer 4 | Opt | Opt | rrect Ansv | mize o | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Single | Beginner | Concept | JEE | Core | L1 | Which of the given statements is/are true? | JDK is a superset of JRE | JVM makes Java platform independent | JVM is highly platform dependent | All of the above | | | 4 | | |
| Single | Expert | Concept | JEE | Core | L1 | Which of the given below interface declaration is applicable only in Java 8? | interface Demo { } | interface Demo { static int method(){ return 1; } default String sayHello(){ return "hello"; } } | interface Demo { public int method() { return 1; } } | interface Demo { static int method(){ return 1; } String sayHello(){ return "hello"; } } | | | 2 | | |
| Single | Beginner | Coding | JEE | Core | L1 | class Shape{ public Shape(){} } class Rectangle extends Shape{ } class MainClass { public static void main(String []args) { Shape shape = new Shape();//line 1 Rectangle rect = (Rectangle)shape;//line 2 } What is the true statement about above code? | This code will compile and run successfully | This will generate ClassCastException as shape can not be typecasted to Rectangle | It will work fine if line 1 be replaced by below given code Shape shape = new Rectangle(); | Both 2 and 3 are true | | | 4 | | |
| Single | Expert | Coding | JEE | util and collections | L1 | public class Employee implements Comparator<Employee>{ int empId; String empName; //getters & setters public Employee(int empId, String empName) { super(); this.empId = empId; this.empName = empName; } @Override public int compare(Employee emp1, Employee emp2) { return emp1.getEmpId()-emp2.getEmpId(); }} public class MainClass { public static void main(String[] args) { TreeSet<Employee>set = new TreeSet<Employee>(); set.add(new Employee(10,"Priya")); set.add(new Employee(9,"Ram")); set.add(new Employee(12,"Riya")); | Employee objects will be sorted as per id | It will generate ClassCastException | It will sort Employee objects as per id if class Employee will implement Comparable and override compareTo method | Both 2 and 3 are true | | | 4 | | |
| Single | Expert | Coding | JEE | util and collections | L1 | import java.util.*; public class ScannerDemo { public static void main(String[] args) { String input = "10 tea 20 coffee 30 tea buiscuits"; Scanner sc = new Scanner(input).useDelimiter("\\s"); System.out.println(sc.next()); System.out.println(sc.next()); System.out.println(sc.next()); System.out.println(sc.next()); sc.close(); } } What will be the output of above code? | 10 tea 20 coffee | It will generate NumberFormatException | tea coffee tea | tea coffee | | | 1 | | |
| Single | Beginner | Concept | JEE | util and collections | | Which of the given method is used to remove element from the list? | remove(int position) | delete(int position) | removeElement(int position) | deleteElement(int position) | | | 1 | | |
| | Expert | Concept | JEE | util and collections | | Which of the given class uses hascode value of an object to determine, how the object should be stored in the collection? | Hashtable | HashMap | ArrayList | Both 1 and 2 | | | 4 | | |
| Single | Expert | Concept | JEE | IO | L1 | class Employee { int empId; String empName; static int count; public Employee(int id, String name) { empId = id; empName = name; count++; } } class MainClass { public static void main(String[] args) { try { FileOutputStream out = new FileOutputStream("d:\\serl.txt"); ObjectOutputStream bout = new ObjectOutputStream(out); Employee e1 = new Employee(111,"RAM"); bout.writeObject(e1); bout.flush(); System.out.println("done"); } catch (FileNotFoundException e) { | It will serialize employee object i.e. empId, empName and count will be store in the file serl.txt | It will serialize only empId and empName | It will generate java.io.NotSerializableException | none of the above | | | 3 | | |

| Q type | Level | Category | Skill | SubSkill | Topic | Q text | Option/ Answer 1 | Option/ Answer 2 | Option/ Answer 3 | Option/ Answer 4 | Opt | Opt | rrect Ans | mize o | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Single | Beginner | Coding | JEE | Stream API | L1 | public class StreamAPIDemo {<br><br> public static void main(String[] args) {<br><br> String []arr = new String[]{"Capgemini","Global", "Solution","Mumbai"};<br>    Stream<String>sarr = Arrays.stream(arr);<br>    String msg = sarr.filter((str)->str.contains("o")).reduce((str1,str2)->str1.concat(str2)).get();<br>        System.out.println(msg);}<br>}<br>What will be the output of above code? | GlobalSolution | Global,Solution | Capgemini,Global,Solution, Mumbai | Global | | | 1 | | |
| Single | Beginner | Coding | JEE | Lambda | L1 | interface CalculateLength<br>{<br> int length(String str);<br>}<br>public class LambdaDemo<br>{<br>public static void main(String []args)<br>{<br>//line1<br>System.out.println(ref.length("capgemini"));<br>}}Which of the given option is correct to insert at line1 for the successful execution of the code ? | CalculateLength ref = (str1)->str1.length(); | int ref = (str1)->str1.length(); | CalculateLength ref = str1->str1.length(); | both 1 and 3 | | | 4 | | |
| Single | Expert | Coding | JEE | JDBC | L1 | public class JDBCdemo {<br><br> public static void main(String[] args) {<br> try {<br>Class.forName("oracle.jdbc.driver.OracleDriver");<br> Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","sys");<br> Statement stmt = con.createStatement();<br> ResultSet rs = stmt.executeQuery("SELECT * FROM Emp");<br> rs.next();<br> while(rs.next())<br> {<br>  System.out.println(rs.getInt(1));<br>  System.out.println(rs.getString(2));<br> }<br> } catch (ClassNotFoundException e) {<br> e.printStackTrace();<br> } catch (SQLException e) {<br> e.printStackTrace();<br> }<br><br><br> }<br><br>} | It will print records from second row onwards | No output | It will throw SQLException as rs.next() is called twice | it will print all records | | | 1 | | |
| Single | Beginner | Concept | JEE | JDBC | L1 | Which of the given statement is used to call stored procedure? | PreparedStatement | CallableStatement | Statement | All of the above | | | 2 | | |
| Single | Expert | Concept | JEE | JDBC | L1 | public class JDBCDemo {<br> public static void main(String[] args) {<br> try {<br>Class.forName("oracle.jdbc.driver.OracleDriver");<br> Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","sys");<br> con.setAutoCommit(false);<br> Statement stmt = con.createStatement();<br> String sql = "INSERT INTO Emp VALUES(119, 'Manager','Sita','Sharma',40000";<br> stmt.executeUpdate(sql);<br> Savepoint sOne = con.setSavepoint();<br> sql = "INSERT INTO Emp VALUES(129, 'Manager','Priti','Verma',50000)";<br> stmt.executeUpdate(sql);<br> Savepoint sTwo = con.setSavepoint();<br> con.rollback(sOne);<br> con.commit();<br> }<br> catch (ClassNotFoundException e) {<br> e.printStackTrace();<br> } catch (SQLException e) {<br> e.printStackTrace();<br> }}}<br>Which of the given statement is true about above code? | It will insert 1 row in the Emp table, with EmpId=119 | It will insert 1 row in the Emp table with EmpId=129 | con.setAutoCommit(false) is not mandatory for the successful execution of this code | No rows will be inserted in the Emp table | | | 1 | | |

| Q type | Level | Category | Skill | SubSkill | Topic | Q text | Option/ Answer 1 | Option/ Answer 2 | Option/ Answer 3 | Option/ Answer 4 | Opt | Opt | Correct Ans | Randomize o | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Single | Expert | Coding | JEE | JDBC | L1 | public class JDBCDemo{<br>public static void main(String[] args) {<br> try {<br>Class.forName("oracle.jdbc.driver.OracleDriver");<br> Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","sys");<br> Statement stmt = con.createStatement();<br> ResultSet rs = stmt.executeQuery("SELECT * FROM EmpDetails");<br> rs.next(); //line 1<br> while(rs.next())// line 2<br> {<br> System.out.println(rs.getInt(1));<br> System.out.println(rs.getString(2));<br> }<br> }<br> catch (ClassNotFoundException e) {<br> e.printStackTrace();<br> } catch (SQLException e) {<br> e.printStackTrace();<br> }}}<br>Assume EmpDetails table has only one row.What will be the output of above code? | No output | It will generate SQLException at line 1 | It will generate SQLException at line 2 | It is print single record | | | 1 | | |
| Single | Beginner | Concept | JEE | JDBC | L1 | Which of the given statement interface supports input output parameters? | Statement | PreparedStatement | CallableStatement | All of the above | | | 3 | | |
| Single | Expert | Concept | JEE | JDBC | L1 | Arrange the below given steps to call procedure?<br>1)Create callable statement object<br>2)Call setXXX() method to set IN parameters<br>3)Call execute() to invoke the procedure/function<br>4)Call getXXX() method to retrieve results from OUT parameters/function return value<br>5)Call registerOutParameter() method to register OUT parameters/function return value | 1,2,5,3,4 | 1,2,3,4,5 | 2,1,3,4,5 | 1,2,4,3,5 | | | 1 | | |
| Single | Beginner | Coding | JEE | Exception | L1 | public class ExceptionDemo {<br><br>public static void main(String[] args) {<br> int i = 0;<br> try<br>    {<br>        i = Integer.parseInt("abc");<br>    }<br> catch(Exception ex)<br>    {<br>        System.out.println("Exception");<br>    }<br> finally<br> {<br> System.out.println("i = "+i);<br> }}}<br>What will be the output of above code? | Exception<br>I = 0 | I = abc | no output | Exception | | | 1 | | |
| Single | Beginner | Concept | JEE | Exception | L1 | Which of the given statement is true about Checked Exceptions? | They are checked by compiler at the time of compilation | They represent exceptions that are frequently considered as "non-fatal" to program execution | IOException, SQLException are checked Exception | All of the above | | | 4 | | |
| Single | Beginner | Coding | JEE | Exception | L1 | public class ExceptionDemo {<br><br>public static void main(String[] args) {<br> try{<br> String str = null;<br> int len = str.length();<br> }<br> catch(Exception ex)<br> {<br> System.out.println("In catch ");<br> throw ex;<br> }}}<br>What will be the output of above code? | In Catch followed by Exception stack trace | In Catch | No Exception | No output | | | 1 | | |
| Single | Beginner | Concept | JEE | Exception | L1 | Which of the given keyword describes the exception, which can be raised by a method? | finally | throw | throws | catch | | | 3 | | |
| Single | Beginner | Concept | JEE | Multithreading | L1 | class MyThread implements Runnable{<br>public void run()<br>{}<br>}<br>class ThreadDemo<br>{<br>public static void main(String []args){<br>Thread thread = new Thread(new Mythread());<br>//line 1<br>}}<br>Which of the given call, when inserted at line 1 will cause the thread to begin the execution? | thread.run() | thread.join() | thread.join(100) | thread.start() | | | 4 | | |
| Single | Beginner | Concept | JEE | Multithreading | L1 | class MyThread implements Runnable{<br>public void run(){<br>System.out.println("Hi");<br>}<br>}<br>public class ThreadDemo<br>{<br> public static void main(String []args){<br>Thread thread = new Thread(new MyThread());<br>thread.start(); //line 1<br>thread.start(); //line 2<br>}<br>}Which of the given is a true statement about above code? | This code will generate output<br>Hi followed by java.lang.IllegalThreadStateException due to line 2 | This code will generate output<br>Hi<br>Hi | This code will generate output<br>Hi | It will generate compilation error as Interrupted Exception is not handled | | | 1 | | |

| Q type | Level | Category | Skill | SubSkill | Topic | Q text | Option/ Answer 1 | Option/ Answer 2 | Option/ Answer 3 | Option/ Answer 4 | Opt | Opt | Correct Answer | Randomize o | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Single | Beginner | Coding | JEE | JUNIT | L1 | `public class Calculator {`<br>`public int add(int num1,int num2)`<br>`{`<br>`return num1+num2;}`<br>`public int div(int num,int den)`<br>`{`<br>`return num/den;}}`<br><br>`public class CalculatorTest {`<br>`Calculator ref;`<br>`@Before`<br>`public void init()`<br>`{`<br>`ref = new Calculator();}`<br>`@Ignore`<br>`@Test`<br>`public void testAdd()`<br>`{`<br>`assertEquals(0,ref.add(3, 4));`<br>`}`<br>`@Test(expected = ArithmeticException.class)`<br>`public void testDiv()`<br>`{`<br>`ref.div(5, 2);`<br>`}}`<br>What will the output after the execution of CalculatorTest? | java.lang.AssertionError for both the test methods | java.lang.AssertionError for testDiv | java.lang.AssertionError for testAdd | Both the test cases will execute succesfuly | | | 2 | | |
| Single | Beginner | Coding | JEE | JUNIT | L1 | `class EmployeeDao`<br>`{`<br>`public Employee insertEmployee(Employee emp)`<br>`{`<br>`//jdbc insert code`<br>`}`<br>`}`<br><br>Which of the given Test class is correct to test the insertEmployee() method and to initialize Dao object only once before all the test case execution? | `class TestEmployeeDao`<br>`{`<br>`EmployeeDao dao;`<br>`@BeforeClass`<br>`public static void init()`<br>`{`<br>`dao = new EmployeeDao();`<br>`}`<br>`@Test`<br>`public void testInsertEmployee(Employee emp)`<br>`{`<br>`emp = dao.insertEmployee(emp);`<br>`assertNotNull(emp);}}` | `class TestEmployeeDao`<br>`{`<br>`EmployeeDao dao;`<br>`@BeforeClass`<br>`public void init()`<br>`{`<br>`dao = new EmployeeDao();`<br>`}`<br>`@Test`<br>`public void testInsertEmployee(Employee emp)`<br>`{`<br>`emp = dao.insertEmployee(emp);`<br>`assertNotNull(emp);}}` | `class TestEmployeeDao`<br>`{`<br>`EmployeeDao dao;`<br>`@Before`<br>`public void init()`<br>`{`<br>`dao = new EmployeeDao();`<br>`}`<br>`@Test`<br>`public void testInsertEmployee(Employee emp)`<br>`{`<br>`emp = dao.insertEmployee(emp);`<br>`assertNotNull(emp);}}` | `class TestEmployeeDao`<br>`{`<br>`EmployeeDao dao;`<br>`@Before`<br>`public static void init()`<br>`{`<br>`dao = new EmployeeDao();`<br>`}`<br>`@Test`<br>`public void testInsertEmployee(Employee emp)`<br>`{`<br>`emp = dao.insertEmployee(emp);`<br>`assertNotNull(emp);}}` | | | 1 | | |
| Single | Expert | Concept | JEE | Log4j | L1 | logger.properties<br><br>log4j.rootLogger=debug,loginfo<br>log4j.appender.loginfo=org.apache.log4j.ConsoleAppender<br>log4j.appender.loginfo.layout=org.apache.log4j.SimpleLayout<br><br>Which of the given statement is true about | rootLogger is assigned with priority level debug and an appender named loginfo, that appender type specified as ConsoleAppender and the appender is assigned a layout SimpleLayout | This logger can log info, debug,warn,fatal,warn and error | This logger can only log debug | Both 1 and 2 | | | 4 | | |
| Single | Expert | Concept | JEE | Log4j | L1 | How will you define the layout of FileAppender using log4j.properties? | log4j.appender.FILE.layout=org.apache.log4j.PatternLayout<br>log4j.appender.FILE.layout.conversionPattern=%m% | log4j.appender.loginfo=org.apache.log4j.ConsoleAppender<br>log4j.appender.loginfo.layout=org.apache.log4j.SimpleLayout | log4j.appender.loginfo=org.apache.log4j.ConsoleAppender<br>log4j.appender.loginfo.layout=org.apache.log4j.HTMLLayout | log4j.appender.loginfo=org.apache.log4j.ConsoleAppender<br>log4j.appender.loginfo.layout=org.apache.log4j.XMLLayout | | | 1 | | |
| Single | Beginner | Coding | JEE | Lambda | L1 | `public class LambdaDemo {`<br><br>`public static void main(String[] args) {`<br>`Supplier<String>supplier = ()->"Welcome to Lambda Expression";`<br>`Predicate<String>pred = (str1)->`<br>`str1.length()>10;`<br>`System.out.println(pred.test(supplier.get()));}}` | TRUE | FALSE | compilation error | none of the above | | | 1 | | |
| Single | Beginner | Coding | JEE | Stream API | L1 | `public class MainClass {`<br>`public static void main(String[] args) {`<br><br>`Integer [] arr = new Integer[]{29,4,45,6,11,14,8,13};`<br><br>`Stream<Integer> sArr = Arrays.stream(arr);`<br>`sArr.limit(6).filter((num)->`<br>`num>4).forEach((num)->System.out.print(num))`<br>`;`<br>`}`<br><br>`}` | 29,45,6,11,14 | 29,4,45,6,11,14 | 29,4,45,6,11,14,8,13 | 29,45,6,11,14,8 | | | 1 | | |
| Single | Expert | Concept | JEE | IO | L1 | `public class FileDemo {`<br><br>`public static void main(String a[]){`<br>`BufferedReader br = null;`<br>`String strLine = "";`<br>`try {`<br>`br = new BufferedReader( new FileReader("d:\\reader.txt"));`<br>`while( (strLine = br.readLine()) != null){`<br>`System.out.println(strLine);`<br>`}`<br>`} catch (FileNotFoundException e) {`<br>`System.err.println("Unable to find the file: fileName");`<br>`} catch (IOException e) {`<br>`System.err.println("Unable to read the file : fileName");`<br>`}`<br>`}`<br>`}`<br>Which of the given statement is true about above code? | It will read the contents of reader.txt line by line | It will read the contents of reader.txt character by character | It will read the contents of reader.txt byte by byte | no output | | | 1 | | |

| Q type | Level | Category | Skill | SubSkill | Topic | Q text | Option/ Answer 1 | Option/ Answer 2 | Option/ Answer 3 | Option/ Answer 4 | Opt | Opt | rrect Answ | mize o | Comments |
|--------|-------|----------|-------|----------|-------|--------|------------------|------------------|------------------|------------------|-----|-----|------------|--------|----------|
| Single | Beginner | Coding | JEE | Core | L1 | class Employee<br>{<br>//getters & setters<br>}<br>class Manager extends Employee<br>{<br>//getters & setters<br>}<br>public class MainClass<br>{<br>public static void main(String []args)<br>{<br>Manager obj = new Manager();<br>System.out.println(obj instanceof Manager);<br>System.out.println(obj instanceof Employee);<br>System.out.println(obj instanceof Object);<br>}<br>}What will be the output of MainClass? | true<br>true<br>true | true<br>false<br>false | true<br>true<br>false | false<br>false<br>false | | | 1 | | |
| Single | Beginner | Concept | JEE | Core | L1 | Which of the given String class method converts data to string? | concat() | substring() | valueOf() | trim() | | | 3 | | |
| Single | Expert | Concept | JEE | Core | L1 | public class LocalDateDemo {<br> public static void main(String[] args) {<br>  DateTimeFormatter format = DateTimeFormatter.ofPattern("dd-MM-yy");<br> String input = "22/09/2017";<br>  LocalDate date = LocalDate.parse(input, format);<br>  System.out.println(date);}}<br>Which of the given statement is true about above code? | It will print date as given below<br>22-09-17 | It will print date as given below<br>22/09/17 | It is will generate DateTimeParseException as format and input are not matching | none of the above | | | 3 | | |
| Single | Expert | Coding | JEE | util and collections | L1 | public class HashMapDemo {<br>public static void main(String[] args) {<br> HashMap<Integer,String> map = new HashMap<Integer,String>();<br> map.put(101, "Capgemini");<br> map.put(102, "Global");<br> map.put(103, "Solutions");<br> Set<Entry<Integer, String>> set = map.entrySet();<br><br> Object [] arr=set.toArray();<br> for( Object val : arr)<br> {<br>  System.out.println(val);<br> }}} | 101=Capgemini<br>102=Global<br>103=Solutions | Capgemini<br>Global<br>Solutions | 101<br>102<br>103 | It will generate Exception | | | 1 | | |
| Single | Expert | Coding | JEE | util and collections | L1 | Hashtable<Integer,String>map = new Hashtable<Integer,String>();<br>Which of the given option is correct to convert map keys into list? | List<Integer>list = new ArrayList<Integer>(map.keySet()); | List<Integer>list = new ArrayList<Integer>(map.entrySet()); | List<Integer>list = new ArrayList<Integer>(map.getKeys()); | List<Integer>list = new ArrayList<Integer>(map.keys()); | | | 1 | | |
| Single | Beginner | Concept | JEE | util and collections | L1 | Implementation of Comparable interface is mandatory to user defined class when the same class objects are added to---- | TreeSet | HashSet | ArrayList | Vector | | | 1 | | |
| Single | Expert | Concept | JEE | util and collections | L1 | ? super T stands for? | any reference types which are superclass of T are allowed. | any reference types which are subclass of T are allowed. | both 1 and 2 | none of the above | | | 1 | | |
| Single | Beginner | Coding | JEE | Lambda | L1 | Which of the given code is correct to accept and print the string and also to compare two numbers using built in functional interface? | Consumer<String>consume = (String str)-> System.out.println(str);<br> consume.accept("Hello World");<br>  BiPredicate<Integer, Integer>predicate =<br>  (num1,num2)->(num1>num2);<br>   System.out.println(predicate.test(45, 2)); | Consumer<String>consume = (String str)-> System.out.println(str);<br> consume.accept("Hello World");<br>  Predicate<Integer, Integer>predicate =<br>  (num1,num2)->(num1>num2);<br>   System.out.println(predicate.test(45, 2)); | Supplier<String>supplier = (String str)-> System.out.println(str);<br> supplier.accept("Hello World");<br>  Predicate<Integer, Integer>predicate =<br>  (num1,num2)->(num1>num2);<br>   System.out.println(predicate.test(45, 2)); | Supplier<String>supplier = (String str)-> System.out.println(str);<br> supplier.accept("Hello World");<br>  BiPredicate<Integer, Integer>predicate =<br>  (num1,num2)->(num1>num2);<br>   System.out.println(predicate.test(45, 2)); | | | 1 | | |
| Single | Beginner | Coding | JEE | Stream API | L1 | public class StreamDemo {<br><br> public static void main(String[] args) {<br>  Stream<String>str = Stream.of("Hello","Hi","Welcome");<br>  str.filter((ref)-> ref.contains("l")).distinct().forEach((str)-> System.out.println(str));<br> }}<br>What is the output of above code? | Hello<br>Welcome | Hello<br>Hi<br>Welcome | Hello<br>Hi | Hi<br>Welcome | | | 1 | | |
| Single | Beginner | Coding | JEE | Lambda | L1 | public class StreamDemo {<br><br> public static void main(String[] args) {<br><br>  BiFunction<String,String,String>fun = (str1, str2)->str1+str2;<br>  String result=null;<br>  result = fun.apply("10", "20");<br>  System.out.println(result);<br> }<br>} | 1020 | 30 | compilation error | Exception | | | 1 | | |