Q.1 Which of the following statements about exception handling in Java is true?
a) Exceptions must always be caught
b) Exceptions cannot be thrown manually
c) Java provides a built-in mechanism to handle exceptions using try-catch blocks
d) Exception handling is not necessary in Java

Ans. c

Q.2 Which keyword is used to throw an exception in Java?
a) try
b) throw
c) throws
d) catch
Ans. b

Q.3 What will happen if an exception is not caught in Java?
a) The program continues execution normally
b) The program terminates abnormally and prints a stack trace
c) The exception is ignored
d) The program prints "Exception occurred" and continues
Ans. b

Q.4 Which exception is thrown when an array index is accessed out of bounds?
a) IOException
b) NullPointerException
c) ArrayIndexOutOfBoundsException
d) IllegalArgumentException
Ans. c

Q.5 Which keyword is used to prevent an object from being serialized?
a) transient
b) static
c) final
d) volatile
Ans. a

Q.6 Which method is used to check if a file exists in Java?
a) exists()
b) isFile()
c) fileFound()
d) isAvailable()
Ans. a
Q.7 What will happen if a file being read does not exist?
a) A new file is created
b) The program continues execution normally
c) A FileNotFoundException is thrown
d) A NullPointerException is thrown
Ans. c

Q.5 Which keyword is used to prevent an object from being serialized?

a) transient
b) static
c) final
d) volatile
Ans. a

The correct answer is **a) transient**.

What is the Transient Keyword?

The `transient` keyword is a modifier that tells Java to ignore an attribute when serializing an object[1]. When a field is marked as `transient`, it prevents that specific field from being serialized during the serialization process[2] [3].

How Transient Works

When an object is serialized, fields marked as `transient` are ignored, and their values are not saved to the file or transmitted over the network[3]. During deserialization, these fields are set to their default values (null for objects, 0 for integers, etc.)[1] [4].

Here's a practical example:

```
class Person implements Serializable {
    String fname = "John";
    String lname = "Doe";
    int age = 24;
    transient int accessCode = 0; // This will not be serialized
}
```

After serialization and deserialization, the `accessCode` field will be reset to its default value (0), while other fields retain their original values[1].

Why Other Options Are Incorrect

b) static - Static fields belong to the class rather than instances and have different behavior during serialization, but they don't prevent serialization of instance fields.

c) final - The `final` keyword makes a variable immutable after initialization but doesn't prevent serialization. You can actually use `transient` and `final` together[3].

d) volatile - The `volatile` keyword ensures thread safety by providing memory visibility in multithreaded environments and has no relation to serialization[4].

Common Use Cases for Transient

Protecting Sensitive Data: Passwords, authentication tokens, or security keys that shouldn't be persisted[3] [5]:

```
private transient String password; // Prevents password from being serialized
```

Excluding Temporary Data: Cache values, intermediate calculations, or runtime-specific data[3]:

```
private transient int tempData; // Temporary data not needed for persistence
private transient Connection dbConnection; // Database connections shouldn't be serialize
```

Performance Optimization: Reducing serialized object size by excluding unnecessary fields[3].

The `transient` keyword is specifically designed for serialization control, making it the only correct answer among the given options for preventing object serialization.

⁂

1. https://www.w3schools.com/java/ref_keyword_transient.asp
2. https://stackoverflow.com/questions/1292474/can-we-deny-a-java-object-from-serialization-other-than-giving-transient-keyword
3. https://unstop.com/blog/transient-keyword-in-java
4. https://www.scaler.com/topics/transient-keyword-in-java/
5. https://www.upgrad.com/blog/transient-keyword-java/

Q.8 Which of the following classes is used for reading and writing bytes to a file?
a) FileWriter
b) FileReader
c) FileInputStream
d) PrintWriter
Ans. c

Q.9 Event class is defined in which of these libraries?
a) java.io
b) java.lang
c) java.net
d) java.util
Ans. d

Q.10 Which method is used to add an element to a collection in java?
a) add()
b) insert()
c) put()
d) push()
ans. a

Q.11 What is the advantage of using generics in Java collections?
a) Generics allow storing only primitive data types
b) Generics enforce type safety at compile time
c) Generics make the code longer and more complex
d) Generics are not supported in Java
Ans. b

Q.12 What does the diamond operator (<>) do in Java generics?
a) It indicates an abstract class
b) It allows automatic type inference
c) It is used only in method declarations
d) It is a required part of all generic classes
Ans. b

Q.13 Which of the following correctly creates a generic ArrayList of Strings?
a) ArrayList<String> list = new ArrayList<>();
b) ArrayList<> list = new ArrayList<String>();
c) ArrayList<String> list = new ArrayList();
d) ArrayList list = new ArrayList<String>();
Ans. a
Q.14 Which collection class maintains the order of insertion and allows duplicate values?
a) HashSet
b) TreeSet
c) ArrayList
d) HashMap
Ans. c

Q.15Which Java collection class maintains elements in a sorted order?
a) HashMap
b) HashSet
c) TreeSet
d) LinkedList
Ans.c

Q.16 Which method adds an element to the front of a Deque?
a) addLast()
b) push()
c) addFirst()
d) enqueue()
Ans. c

Q.17 Which of the following statements about generics is true?
a) Generics can only be used with collections
b) Generics prevent runtime ClassCastException
c) Generics cannot be used with interfaces
d) Generics allow multiple types in a single class
Ans. b

Q.18 Which of the following statements is true about assertions in Java?
a) Assertions are used to catch runtime exceptions
b) Assertions are used for debugging and checking invariants
c) Assertions should be used for handling user inputs
d) Assertions replace exception handling
Ans. b

Q19. What is the output for the below code ?

```
interface Demo{
    public void Value();
}

1. public class Test{
2.      public static void main (String[] args){
3.            Demo d = new Demo(){
4.                         public void printValue(){
5.                               System.out.println("A");
6.                         }
7.                   };
8.            d.Value();
9.      }
10. }
```

A.Compilation fails due to an error on line 3
B.A
C.Compilation fails due to an error on line 8

D.null
ans. A

Q20. What will happen when you compile and run the following code?

```
class DemoInner{
        public void print(){
                System.out.println("Hi");
        }
}

public class Demo
{
        public static void main(String[] args)
         {
                DemoInner t = new DemoInner()
                {
                        public void print(){
                                System.out.println("Hello");
                        }
                };

                t.print();
        }
}
```

1. Hi
2. Hello
3. Compilation error
4. No output

ans 2

Q.21
Determine output of the following code.

```
interface x { }

class z { }

class y extends z { }

class w extends y implements x { }

public class Test extends Thread{
      public static void main(String[] args){
            w b = new w();
            if (b instanceof x)
                  System.out.println("b is an instance of x");
```

```
        if (b instanceof z)
            System.out.println("b is an instance of z");
    }
}
```

A.Nothing.
B.b is an instance of A.
C.b is an instance of C.
D.b is an instance of A followed by b is an instance of C.

ans d

```
Q.22 class Error1{
   public static void main(String args[]){
    int a = 10;
    int b = 5;
    int c = 5;
    try{
    x = a / (b-c);
    catch (AtrithmeticException e){
      System.out.println("Division by zero");
    }
    y = a / (b+c);
    System.out.println("y = "+y);
   }
}
```

```
Q23. import java.util.*;
public class Demo {
  public static void main(String[] args) {
    TreeSet<String> ts = new TreeSet<String>();
    ts.add("X");
    ts.add("X");
    System.out.println(ts);
  }
}
```

Q24. Which of the following option can be used at MISSING-1, it won’t raise compile-time error or Exception?

```
import java.util.*;
public class Test {
  public static void main m1(ArrayList<?> al)  {
    // MISSING-1
  }
}
```

a) al.add(“A”);
b)al.add(10);
c) al.add(null);
d) None of these
ans. C

Q25. Identify the correct characteristic of an abstract class in Java.

a) It cannot have constructors

b) It must contain at least one abstract method

c) It cannot have final methods

d) It must be instantiated directly

Answer: b) It must contain at least one abstract method

Q26. Identify the purpose of using the final keyword in Java.

a) To allow method overriding

b) To restrict modification of variables, methods, or classes

c) To enable multiple inheritance

d) To define an abstract method

Answer: b) To restrict modification of variables, methods, or classes

Q27. Identify the difference between top-level and nested classes.

a) A nested class can exist independently of its enclosing class

b) A top-level class must be declared as static

c) A nested class is declared within another class

d) A top-level class cannot have member variables

Answer: c) A nested class is declared within another class

Q28. Identify the correct statement about Java interfaces.

a) Interfaces cannot have default methods

b) An interface can extend multiple classes

c) Interfaces can have both abstract and default methods

d) An interface must have a constructor

Answer: c) Interfaces can have both abstract and default methods

Q29. Identify which feature allows objects of different classes to be treated as instances of the same type.

a) Encapsulation

b) Inheritance

c) Polymorphism

d) Abstraction

Answer: c) Polymorphism

Q30. Identify the correct syntax for defining a lambda expression.

a) (int a, int b) -> { return a + b; }

b) lambda (int a, int b) { return a + b; }

c) (int a, int b) { return a + b; }

d) new lambda (int a, int b) { return a + b; }

Answer: a) (int a, int b) -> { return a + b; }

Q31. Identify the correct statement about anonymous inner classes.

a) They must implement at least two interfaces

b) They cannot extend a class

c) They are used for defining one-time-use classes

d) They must be static

Answer: c) They are used for defining one-time-use classes

Q32. Identify the correct way to extend a Java interface.

a) class MyInterface extends AnotherInterface {}

b) interface MyInterface extends AnotherInterface {}

c) interface MyInterface implements AnotherInterface {}

d) class MyInterface implements AnotherInterface {}

Answer: b) interface MyInterface extends AnotherInterface {}

Q33. Identify the advantage of using the List interface in Java.

a) It only supports arrays of fixed size0

b) It provides dynamic resizing and ordered elements

c) It does not support generics

d) It does not allow duplicate elements

Answer: b) It provides dynamic resizing and ordered elements

Q34. Identify the correct feature of default methods in interfaces.

a) They must be overridden in implementing classes

b) They can have a body and provide default behavior

c) They cannot be static

d) They must be declared as abstract

Answer: b) They can have a body and provide default behavior

Theory Qns.

Q.1 What is the Hashtable class in Java?Write a Java program to demonstrate how to create and use a Hashtable.
Q.2 What is a collection in Java? Explain its significance.
Q.3 Explain how exception handling mechanishm can be used for debugging a program. Give a suitable example.

Q.4 Distinguish between
a) InputStream and Reader classes
b) OutputStream and Writer classes
Q.5 Explain the concept of an interface in Java. How is it different from an abstract class?
Q.6 Discuss the concept of classes and objects in java
Q.7 Write a Java program to create a class called Shape with a method called getArea(). Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.
Q.8 Explain exception handling in java. Write a program for the same. Also differentiate between error and exception.
Q.9 What is constructor? Discuss the types of constructor with program with proper explanation.
Q10. Compare String and StringBuilder classes of java. In what scenarios would you prefer StringBuilder over String? Justify your answer with examples.