1. When does Exceptions in Java arises in code sequence?
   1. Run Time b) Compilation Time c) Can Occur Any Time d) None of the mentioned
2. Which of these keywords is not a part of exception handling? a) try

b) finally c) thrown d) catch

1. Which of these keywords must be used to monitor for exceptions?
   1. try b) finally c) throw d) catch
2. Which of the following are the most common run-time errors in Java programming.
3. Missing semicolons
4. Dividing an integer by zero
5. Converting invalid string to number
6. Which of the following are the most common compile time errors in Java programming.
7. Missing semicolons
8. Use of undeclared variables
9. Attempting to use a negative size for an array
10. Bad reference of objects
    1. i, ii and iii only
    2. ii, iii and iv only
    3. i, ii and iv only
    4. All i, ii, iii and iv
11. The unexpected situations that may occur during program execution are
12. Running out of memory
13. Resource allocation errors
14. Inability to find a file
15. Problems in network
    1. i, ii and iii only
    2. ii, iii and iv only
    3. i, ii and iv only
    4. All i, ii, iii and iv
16. The class at the top of the exception classes hierarchy is called ……………………
17. throwable
18. catchable
19. hierarchical
20. ArrayIndexOutofBounds
21. ………………… exception is thrown when an exceptional arithmetic condition has occurred.
22. Numerical
23. Arithmetic
24. Mathematical
25. All of the above
26. ………………….. exception is thrown when an attempt is made to access an array element beyond the index of the array.
27. Throwable
28. Restricted
29. Security
30. ArrayIndexOutofBounds
31. You can implement exception-handling in your program by using which of the following keywords.

i) Try ii) NestTry iii) Catch iv) Finally

1. i, ii and iii only
2. ii, iii and iv only
3. i, iii and iv only
4. All i, ii, iii and iv
5. The ……………………. statement is passed a single parameter, which is reference to the exception object thrown.
6. throw
7. catch
8. finally
9. try
10. Every try statement should be followed by at least one catch statement; otherwise …………………. will occur.
11. no execution
12. null
13. zero
14. compilation error
15. If an exception occurs within the …………………….. block, the appropriate exception-handler that is associated with the try block handles the exception.
16. throw
17. catch
18. finally
19. try
20. …………………….. is caused by general I/O failures, such as inability to read from file.
21. I/O failure
22. I/O exception
23. I/O inability
24. I/O distortion
25. What will be the output of the program?

public class Foo

{

public static void main(String[] args)

{

try

{

return;

}

finally

{

System.out.println( "Finally" );

}

}

}

1. Finally
2. Compilation fails.
3. The code runs with no output.
4. An exception is thrown at runtime.
5. What will be the output of the program?

try

{

int x = 0; int y = 5 / x;

}

catch (Exception e)

{

System.out.println("Exception");

}

catch (ArithmeticException ae)

{

System.out.println(" Arithmetic Exception");

}

System.out.println("finished");

1. finished
2. Exception
3. Compilation fails.
4. Arithmetic Exception
5. What will be the output of the program?

public class X

{

public static void main(String [] args)

{

try

{

badMethod(); System.out.print("A");

}

catch (RuntimeException ex) /\* Line 10 \*/

{

System.out.print("B");

}

catch (Exception ex1)

{

System.out.print("C");

}

finally

{

System.out.print("D");

}

System.out.print("E");

}

public static void badMethod()

{

throw new RuntimeException();

}

}

1. BD
2. BCD
3. BDE
4. BCDE
5. What will be the output of the program?

public class Test

{

public static void aMethod() throws Exception

{

try /\* Line 5 \*/

{

throw new Exception(); /\* Line 7 \*/

}

finally /\* Line 9 \*/

{

System.out.print("finally "); /\* Line 11 \*/

}

}

public static void main(String args[])

{

try

{

aMethod();

}

catch (Exception e) /\* Line 20 \*/

{

System.out.print("exception ");

}

System.out.print("finished"); /\* Line 24 \*/

}

}

1. finally
2. exception finished
3. finally exception finished
4. Compilation fails
5. What will be the output of the program?

public class MyProgram

{

public static void main(String args[])

{

try

{

System.out.print("Hello world ");

}

finally

{

System.out.println("Finally executing ");

}

}

}

1. Nothing. The program will not compile because no exceptions are specified.
2. Nothing. The program will not compile because no catch clauses are specified.
3. Hello world.
4. Hello world Finally executing 20.What will be the output of the program?

class Exc0 extends Exception { }

class Exc1 extends Exc0 { } /\* Line 2 \*/ public class Test

{

public static void main(String args[])

{

try

{

throw new Exc1(); /\* Line 9 \*/

}

catch (Exc0 e0) /\* Line 11 \*/

{

System.out.println("Ex0 caught");

}

catch (Exception e)

{

System.out.println("exception caught");

}

}

}

1. Ex0 caught
2. exception caught
3. Compilation fails because of an error at line 2.
4. Compilation fails because of an error at line 9.