EXCEPTIONS

Viva Voce on Exceptions

Q1. What is the difference between error and exception in java?

Errors are mainly caused by the environment in which an application is running. For example, OutOfMemoryError happens when JVM runs out of memory.

Exceptions are mainly caused by the application itself. For example, NullPointerException occurs when an application tries to access null object.

Q2. Can we write only try block without catch and finally blocks?

No, It shows compilation error. The try block must be followed by either catch or finally block. You can remove either catch block or finally block but not both.

Q3. What are run time exceptions in java. Give example?

The exceptions which occur at run time. These exceptions are unknown to compiler. All sub classes of java.lang.RunTimeException and java.lang.Error are run time exceptions. These exceptions are unchecked type of exceptions. For example,

- NumberFormatException,
- NullPointerException,
- ClassCastException,
- ArrayIndexOutOfBoundException,
- StackOverflowError etc.

Q4. What is OutOfMemoryError in java?

OutOfMemoryError is the sub class of java.lang.Error which occurs when JVM runs out of memory.

Q5. Can we keep the statements after finally block If the control is returning from the finally block itself?

No, it gives unreachable code error. Because, control is returning from the finally block itself.

Compiler will not see the statements after it. That's why it shows unreachable code error

Q6. Does finally block get executed If either try or catch blocks are returning the control?

Yes.

Q7. What is Re-throwing an exception in java?

Exceptions raised in the try block are handled in the catch block. If it is unable to handle that exception, it can re-throw that exception using throw keyword. It is called re-throwing an exception.

Q8. What is ClassCastException in java?

ClassCastException is a RunTimeException which occurs when JVM unable to cast an object of one type to another type.

Q9. Which class is the super class for all types of errors and exceptions in java?

java.lang.Throwable is the super class for all types of errors and exceptions in java.

Q10. What is the use of printStackTrace() method?

printStackTrace() method is used to print the detailed information about the exception occurred.

Q11. Give some examples to checked exceptions?

ClassNotFoundException, SQLException, IOException.

Q12. Give some examples to unchecked exceptions?

NullPointerException, ArrayIndexOutOfBoundsException, NumberFormatException.

Q13. What Is the Difference Between Throw and Throws Keywords in Java?

The throws keyword is used with a method signature to declare exceptions that the method might throw, whereas the throw keyword is used to disrupt the flow of a program and handing over the exception object to run-time to handle it.

Q14. What Happens When an Exception Is Thrown by the Main Method?

When an exception is thrown by mainmethod(), Java Runtime terminates the program and prints the exception message and the stack trace in-system console.

Q15. What Is a Stacktrace and How Does it Relate to an Exception?

A stack trace provides the names of the classes and methods that were called, from the start of the application to the point that an exception occurred. It's a very useful debugging tool since it enables us to determine exactly where the exception was thrown in the application and the original causes that led to it.

Q16. What is the difference between a checked and an unchecked exception?

A checked exception must be handled within a try-catch block or declared in a throws clause; whereas an unchecked exception is not required to be handled nor declared. Checked and unchecked exceptions are also known as compile-time and runtime exceptions respectively. All exceptions are checked exceptions, except those indicated by Error, RuntimeException, and their subclasses.

MCQs on Exceptions

Q1. When does Exceptions in Java arises in code sequence?

- A. Run Time
- B. Compilation Time
- C. Can Occur Any Time

Q2. Which of these keywords is not a part of exception handling?

- A. try
- B. finally
- C. thrown

Q3. Which of these keywords must be used to monitor for exceptions?

- A. try
- B. finally
- C. throw

Q4. Which of these keywords is used to manually throw an exception?

● A. try

● B. finally

● C. throw

Q5. Which of these is a super class of all exceptional type classes?

● A. String

● B. RuntimeExceptions

● C. Throwable

Q6. Which of these class is related to all the exceptions that cannot be caught?

● A. Error

● B. Exception

● C. RuntimeExecption

Q7. Which of these handles the exception when no catch is used?

● A. Default handler

● B. finally

● C. throw handler

MCQs Answers

Q1. (A)

Q2. (C)

Q3. (A)

Q4. (C)

Q5. (C)

Q6. (B)

Q7. (C)

References:

1. Sebesta,"Concept of programming Language", Pearson Edu

2. Louden, “Programming Languages: Principles & Practices” , Cengage Learning
3. Tucker, “Programming Languages: Principles and paradigms “, Tata McGraw –Hill.
4. E Horowitz, “Programming Languages”, 2nd Edition, Addison Wesley

Related posts:

1. Sequence Control & Expression | PPL
2. PPL:Named Constants
3. Parse Tree | PPL | Prof. Jayesh Umre
4. Basic elements of Prolog
5. Loops | PPL | Prof. Jayesh Umre
6. Subprograms Parameter passing methods | PPL | Prof. Jayesh Umre
7. Programming Paradigms | PPL | Prof. Jayesh Umre
8. Subprograms Introduction | PPL | Prof. Jayesh Umre
9. Phases of Compiler | PPL | Prof. Jayesh Umre
10. Parse Tree | PPL
11. Influences on Language design | PPL | Prof. Jayesh Umre
12. Fundamentals of Subprograms | PPL | Prof. Jayesh Umre
13. Programming Paradigm
14. Influences on Language Design
15. Language Evaluation Criteria
16. OOP in C++ | PPL
17. OOP in C# | PPL
18. OOP in Java | PPL
19. PPL: Abstraction & Encapsulation
20. PPL: Semaphores
21. PPL: Introduction to 4GL

22. PPL: Variable Initialization
23. PPL: Conditional Statements
24. PPL: Array
25. PPL: Strong Typing
26. PPL: Coroutines
27. PPL: Exception Handler in C++
28. PPL: OOP in PHP
29. PPL: Character Data Type
30. PPL: Heap based storage management
31. PPL: Primitive Data Type
32. PPL: Data types
33. Programming Environments | PPL
34. Virtual Machine | PPL
35. PPL: Local referencing environments
36. Generic Subprograms
37. Local referencing environments | PPL | Prof. Jayesh Umre
38. Generic Subprograms | PPL | Prof. Jayesh Umre
39. PPL: Java Threads
40. PPL: Loops
41. PPL: Exception Handling
42. PPL: C# Threads
43. Pointer & Reference Type | PPL
44. Scope and lifetime of variable
45. Design issues for functions
46. Parameter passing methods
47. Fundamentals of sub-programs
48. Subprograms

49. Design issues of subprogram
50. Garbage Collection
51. Issues in Language Translation
52. PPL Previous years solved papers
53. Type Checking | PPL | Prof. Jayesh Umre
54. PPL RGPV May 2018 solved paper discussion| Prof. Jayesh Umre
55. PPL Viva Voce
56. PPL RGPV June 2017 Solved paper | Prof. Jayesh Umre
57. Concurrency
58. Basic elements of Prolog
59. Introduction and overview of Logic programming
60. Application of Logic programming
61. PPL: Influences on Language Design
62. Language Evaluation Criteria PPL
63. PPL: Sequence Control & Expression
64. PPL: Programming Environments
65. PPL: Virtual Machine
66. PPL: Programming Paradigm
67. PPL: Pointer & Reference Type
68. try-catch block in C++