|  |  |
| --- | --- |
| 1. | **What happens if while executing a java program if any statement is producing abnormal condition?**  Corresponding exception or error object is raised or created. |
| 2. | **What happens if while executing a java program if any statement is producing abnormal condition and it is not handled?**  Corresponding exception or error object is raised or created. If that exception or error object is not handled the flow will be terminating. In order, to handle exception/error object to continue the flow we have to use Exception Handling. |
| 3. | **What is exception handling?**  Mechanism used to handle raised exception/error object is called as Exception Handling. |
| 4. | While the flow of execution terminates due to exception/error object does it provide any information? Specify the points described in the information.  Yes it do provides information.  There are 4 important points in that information:   1. In which thread exception/error has occurred 2. The exception class i.e. Name of the exception 3. Reason: Why we are getting an exception? It is optional 4. Exactly at which statement exception is raised.   E.g. : Exception in thread “main” java.lang.ArithmeticException: / by zero  at com.lara.A.main(A.java : 7)  1 2 3 4 |
| 5. | **State scenario when a specific exception/error will be raised.**  **java.lang.ArithmeticException** :- While dividing a number by zero.  **java.lang.NullPointerException** :- Calling a member from null reference.  **java.lang.NumberFormatException** :- Unable to format a String into a number.  **java.lang.StackOverflowError** :- While making a recursive call to method.  **java.lang.OutOfMemoryError**:- Creating an array with size which is not possible to accommodate in available heap. |
| 6. | **Which is the super class for Exception and Error?**  Throwable |
| 7. | **What output you get on printing any exception object i.e. System.out.println(ex)?**  ExceptionClassName:reason |
| 8. | **What output you get on printing any exception object getMessage() i.e. System.out.println(ex.getMessage())?**  It prints only reason for exception. |
| 9. | **What does ex.printStackTrace() prints?**   1. It prints The exception class i.e. Name of the exception 2. Reason: Why we are getting an exception? It is optional 3. Exactly at which statement exception is raised. |
| 10. | **Amongst 3 different ways to print an exception message which is advisable?**  ex.printStackTrace() |
| 11. | **If any local variable is available in try block can it be used in catch block too?**  No. local variable is specific to a particular block only. |
| 12. | **If you want to use any local variable inside try and catch block where should it be declared?**  Before try block. |
| 13. | **Explain try..catch.**  Inside java program if you identify statement which might produce abnormal condition / error-prone statements place it in try block.  Inside catch block keep the statements which should be executing to handle the exception.  catch will not be executing by default, whenever exception object is occurring to handle that catch will be executing. |
| 14. | **What if an exception occurs in catch block?**  If an exception occurs in catch block it checks whether the error-prone statements are defined in the try block and there is a catch block defined within it to handle the exception or not. If it is defined then will handle the exception otherwise will terminate the execution of program. |
| 15. | **Can we have nested try..catch within the try block or catch block?**  Yes we can have nested try.. catch block within try or catch block. |
| 16. | **Can there be multiple catch block for the same try block?**  Yes. We can have multiple catch block for a single try block.  Points to be remembered while specifying multiple catch block for try block:-   1. While keeping multiple catch, every catch should take different exception type as arg. 2. Whenever exception is occurring inside try maximum 1 catch block will be executing. In no scenario more than one catch block will be executing. 3. If an exception occurs it checks from top to bottom for matching catch block with compatible exception type. If it finds one then executes that particular catch block. Otherwise, it doesn’t execute any catch block 4. If no exception occurs then too no catch will be executed. |
| 17. | **What is the output of the following code ?**  class ExceptionHandling {  public static void main(String args[]) {  try {  int a, b;  b = 0;  a = 5 / b;  System.out.print("A");  }  catch(ArithmeticException e) {  System.out.print("B");  }  }  }  **Output :**  B |
| 18. | **Which keyword is used to monitor statement for exception?**  try |
| 19. | **If an exception is generated in try block , then it is caught in \_\_\_\_ block.**  catch |
| 20. | **What is the purpose of finally block?**  finally block will be executing in every execution. No matter whether exception is raising or not. No matter whether raised exception object is handled or not. No matter whether return statement is there or not. |
| 21. | **What if any exception or error occurs before try block, whether finally block will be executed?**  If an abnormal condition occurs before try block then finally block will not be executed in any case. |
| 22. | **If an exception occurs in catch block, whether flow will be directly terminated or finally block will be executed?**  If an exception occurs in the catch block flow is supposed to be terminated but before terminating finally block will be executed. |
| 23. | **In which scenario finally block will not be executed?**  If there is System.exit(0) is used in try block then finally block will not be executed because it has to terminate the execution of program. |
| 24. | **What is an exception?**  In Java program, if any statement is producing abnormal condition for that abnormal condition, an exception object is being created. |
| 25. | **What is an error?**  While running java program an abnormal condition is occurring due to external resources for that an error object will be created.  It can be memory management, required class isn’t available, required method isn’t available. |
| 26. | **Tell something about checked exception.**  Compiler is having every information about the checked exception.  Compiler forces you to place in try catch whenever it is required.  If try..catch isn’t required and programmer has placed it will result in compile time error. |
| 27. | **Tell something about unchecked exception.**  Compiler doesn’t have information or knowledge about particular type of error or exception.  Compiler isn’t forcing you to keep try..catch. Without try..catch also we can compile.  Compiler is least bother about unchecked exception. |
| 28. | **Which subclasses are unchecked exception?**  Error and RuntimeException. |
| 29. | **Amongst IOException and FileNotFoundException both are checked exception which one is superclass?**  IOException |
| 30 | **If I place System.out.println() between 2 catch blocks. What happens?**  We can’t incorporate any statement between 2 catch blocks. It will result in compile time error. |
| 31. | **If I place System.out.println() after try block and before catch block. What happens?**  We can’t place any statement between try block and catch block. It will result in compile time error. |
| 32. | **Can there be a try block without a catch/finally block?**  No. |
| 33. | **When an array element is accessed beyond the array size , \_\_\_\_ exception is thrown.**  ArrayIndexOutOfBoundsException |
| 34. | **parseInt(), parseLong() throws which type of exception?**  NumberFormatException |
| 35. | **The classes that extend runtime exception throw exception called**  Unchecked Exception |
| 36. | **Can a same try block have possibility to throw multiple exceptions in single execution?**  No. |
| 37. | **Can I use more than 1 try block in single program?**  Yes. One can place any no. of try block in single program. |
| 38. | **If there is a method named test()which has an unhandled exception, being called from test().How unhandled exception will be propogated?**  Unhandled exception object is returned to caller of test() since it is not handled in test().i.e. main(). Now in main(), if there is code to handle exception then it will be handled. Otherwise, unhandled exception object will be returned to its caller i.e. java command. When java command is receiving the exception it prints 4 details of occurred exception. |
| 39. | **Exception Hierarchy**  http://2.bp.blogspot.com/-MvJZdGu9pLw/UlzeUNxvjwI/AAAAAAAAAwk/LlyzVBDX0L8/s1600/exceptions.png |
| 40. | **If we have an empty try block followed by a catch with any checked exception. Will it compile?**  No. As there is no statement in try block with possibility of the specified checked exception so it won’t compile. |
| 41. | **If we have java statement with possibility of checked exception but we have not placed it in try block with appropriate catch. Will it compile?**  No. As there is a statement with possibility of checked exception it should be placed in try block and there should be a catch block which can handle that checked exception. |
| 42. | **Which exception is possible for Class.forName(“”)? Is it checked or unchecked exception?**  It has possibility to throw ClassNotFoundException. It is a checked exception. |
| 43. | **What is the purpose of throws clause?**  In case of checked exception if you are not placing try..catch then compiler will be looking for throws.  throws is only must for checked exception.  Through throws you can delegate/propogate/assign the exception to the caller to handle.  If you are not mentioning throws then try with appropriate checked exception is mandatory. |
| 44. | **If a method test() contains statement which has possibility of checked exception, called from main(). If it is not placed in try block with appropriate catch. While compiling what will be compiler looking for?**  test() should have throws statement with the possible checked exception. As well as main() should call test() either in try block with same checked exception i.e. thrown from the test() catch block. Or main() should also have throws statement specifying the same checked exception |
| 45. | **If I have an empty method which throws a checked exception will it compile fine?**  Yes. Whether a method contains a statement with possibility of the specified checked exception or not it can have throws with that checked exception. |
| 46. | **For exception’s purpose throws or catch argument can be?**  Any subclass of Throwable |
| 47. | **If test() method has possibility to throw SQLException, IOException. Can we specify throws Exception rather than specifying SQLException, IOException?**  Yes. It can be specified as both are the subclasses to Exception, both can be upcasted to super class. i.e. Exception. |
| 48. | **While overloading a constructor if no arg constructor is throwing checked exception and it is called from other constructor then to handle the exception. What can be done?**  One need to use throws checked exception. As in constructor 1st statement should be super or this calling statement. So no possibility for using try..catch. |
| 49. | **If a class A contains no arg constructor that throws checked exception. Class B is inheriting class A containing no arg constructor. What is required to compile the code successfully?**  constructor in class B i.e. no arg constructor must also have throws checked exception same as class A constructor. |
| 50. | **Is it possible to keep multiple catch for same try?**  Yes.  If all catch are taking exception in same inheritance then it should be most specific to most generic.  No more than 1 catch should have same exception as argument.  Maximum 1 catch will only be executing even though you are placing multiple catch. |
| 51. | **Certain rules with regards to try, catch, finally.**  1 try can be with :   1. 1 catch 2. Multiple catch 3. 1 catch, 1 finally 4. Multiple catch, 1 finally 5. 1 finally   You can’t keep catch alone.  You can’t keep finally alone.  try should be preceeding to catch, finally.  finally should be always the last block when used along with try followed by catch. |
| 52. | **In which scenario we can have try..finally?**  We are not handling exception but some mandatory statements will be executed whether exception occurs or not. |
| 53. | **Which package contains exception handling related classes?**  java.lang |
| 54. | **The subclass exception should precede the base class exception when used within the catch clause. True/False?**  True |
| 55. | **Exceptions can be caught or rethrown to a calling method. True/False?**  True |
| 56. | **When a program does not want to handle exception, the \_\_\_\_\_\_ is used.**  Throws |