Java – Learn From Home

Assignment – Chapter 4

Concept: Exception-Handling

Objective: At the end of the assignment, participants will be able to:

- Understand How to use try, catch, throw, throws, finally
- How to propagate Exceptions
- How to Create User Defined Exceptions

Problems:

Exercise 1:

1. Write a program to demonstrate the use of try, catch, finally throw and throws keywords and demonstrate the following points in the program.
   a) Multiple catch blocks.
   b) try-catch-finally combination.
   c) try-finally combination.
   d) Exception propagation among many methods.
   e) Use of getMessage(), printStackTrace() function of Throwable class.
   f) Nested try blocks

Guided Solution:

Step 1: Create an ExceptionHandling class, in which we will have main method.

Step 2: Inside main method declare a string variable name and initialize with null.

Step 3: provide outer try block in which take an inner try block, and raise explicitly ArithmeticExcepton by writing s.o.p(10/0), where you are not handling it by inner catch, and propagating that exception to the outer try catch block.Inside catch block use ae.printstacktrace() to print entire exception information and ae.getmessage() is used to print only the description of the exception

Step 4: And finally finally block will be executed.

Exercise 2:

2. Write a program to throw a checked exception explicitly using 'throw' keyword and

a) Handle the exception in same method.

b) use throws clause and handle the exception in some other method (calling method)

c) Don't either handle or use the throws clause.

Guided Solution:

Step 1: Create a class ThrowsException, inside this class create a main method.

Step 2: Provide a try block in which explicitly create a checked exception using throw new Exception() and handle it with try catch block.

Step 3: Inside main method call doStuff(), in which call doMoreStuff(), inside this method raise explicitly a checked exception by specifying throw new Exception() and don't handle it with either try catch or throws keyword , simply leave it, which finally causes compile time error

Step 4: After calling doMoreStuff() , then again raise explicitly exception by specifying throw new Exception(), now delegate the exception handling responsibility to the caller of this doStuff() which is main method by specifying throws after doStuff() .

Exercise 3:

3. Create a user defined exception to check whether your employee exist in your data structure (use any data structure to store the employees - like array, ArrayList etc.) and throw exception if name is not in the employees list. Use the catch and finally block to make an appropriate)

Guided Solution:

Step 1: create an Employee class and provide instance variables like ename,empid.
Step 2: Using Employee class constructor initialize ename, and empid.
Step 3: And provide getter method for each instance variable
Step 4: create a class EmployeeNotFoundExcepton which extends Exception class
Step 5: Inside this class create a constructor , which is used to pass exception description to the parent class i.e Exception class and initialize this description using parent class constructor.
Step 6: create a Test class in which we will create a main method ,inside main method create scanner class object using this object we will accept ename from the user.
Step 7: And then create ArrayList object.
Step 8: Now create three employee with different names and ids.
Step 9: And these objects to the ArrayList
Step 10: Using Iterator traverse all the employee objects from the ArrayList. While traversing check if the employee exists or not , if exists display employee name and id otherwise throws EmployeeNotFoundException