Object Oriented Programming with JAVA
Lab -6 Exception Handling and Packages

Core

1 Write a program to create a queue with two operations : Dequeue and Enqueue. Write custom exception classes UnderFlowException and OverFlowException to indicate underflow and overflow of queue respectively(override toString() method in both classes). Write dequeue and enqueue methods such that when underflow or overflow occur, it will throw appropriate custom exception.

Write a DemoCustomException class which contain main method to handle CustomException (i.e. UnderFlowException and OverFlowException)

2 Write a program to demonstrate the use of chained exception

3 Consider Workdir and Sourcedir are the directories on the desktop. Consider A.java, B.java, C.java, D.java and Demo.java file written inside Sourcedir. Corresponding class files are in packages pack1, pack2, pack1.p, pack3 and pack4 under the workdir. The classes A,B,C,D have constructors and dispA, dispB, dispC, dispD methods respectively which prints necessary details. Demo class uses the other classes, call constructors and methods of other classes. Write a program to do so. Specify import statements at necessary places also set classpath whenever required.

Plus

1 Write a program that demonstrates the use of **throws** keyword for propagation of checked exception

- Create a class Simple with three methods m(), n(), p() with no arguments and void return type. m() and n() throws IOException. p() uses try-catch block.
- In m(), throw new IOException("device error") as a checked exception. Call m() from n(). Call n() from p().
- In the main(), create a Simple class object and call the p() method.

2 Write a program that creates and throws custom exception using **throw** keyword. Handle the custom exception in your program and print it.

- Create a class **InvalidAgeException** that extends **Exception**. The constructor of **InvalidAgeException** with a **String** argument calls superclass constructor.
- Create a **CustomExceptionDemo** class that has a static method void **validate** that take integer argument age and returns void. This method **throws InvalidAgeException**. Throw a **new InvalidAgeException** if the age is below 18 years. From the main method, call the validate method.

3 Write a program which shows where the access of public, protected, default and private member is possible. Create at least two packages & necessary classes, subclasses between packages to show the use of “protected” and “public” modifier