**EX. NO: 8 EXCEPTION HANDLING**

**(27/3/2017-2/4/2017)**

**Note: Part I and Part 2 should be recorded in your observation as directed by your mentor**

**Part-I (Who should do Part-I?)**

***Anyone who wants to clear java, hope everyone wants to…….***

**7. a)** Write a Java program with proper naming conventions and comments to illustrate the concept of: **try, catch, throw, throws** and **finally**

**7. b)** Write a Java program for user defined exception that checks the internal and external marks; if the internal marks is greater than 30 it raises the exception “Internal mark exceeded”; if the external marks is greater than 70 it raises the exception and displays the message “External mark exceeded”, Create the above exception and test the exceptions.

**Part-II: MCQ (Who should do Part-II?)**

***Who will not love fun? If you love fun go ahead …and you can take this fun back home and continue there as well……………….***

1.What will be the output of the program?

public class Foo

{

public static void main(String[] args)

{

try

{

return;

}

finally

{

System.out.println( "Finally" );

}

}

}

A. Finally

B. Compilation fails.

C. The code runs with no output.

D. An exception is thrown at runtime.

2. 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");

A. finished B. Exception

C. Compilation fails D. Arithmetic Exception

3. What will be the output of the program?

public class X

{

public static void main(String [] args)

{

try

{

badMethod();

System.out.print("A");

}

catch (Exception ex)

{

System.out.print("B");

}

finally

{

System.out.print("C");

}

System.out.print("D");

}

public static void badMethod()

{

throw new Error(); /\* Line 22 \*/

}

}

A. ABCD

B. Compilation fails.

C. C is printed before exiting with an error message.

D. BC is printed before exiting with an error message.

4. 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();

}

}

A. BD B. BCD

C. BDE D. BCDE

5. What will be the output of the program?

public class RTExcept

{

public static void throwit ()

{

System.out.print("throwit ");

throw new RuntimeException();

}

public static void main(String [] args)

{

try

{

System.out.print("hello ");

throwit();

}

catch (Exception re )

{

System.out.print("caught ");

}

finally

{

System.out.print("finally ");

}

System.out.println("after ");

}

}

A. hello throwit caught

B. Compilation fails

C. hello throwit RuntimeException caught after

D. hello throwit caught finally after

6. 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 \*/

}

}

A. finally

B. exception finished

C. finally exception finished

D. Compilation fails

7. What will be the output of the program?

public class X

{

public static void main(String [] args)

{

try

{

badMethod();

System.out.print("A");

}

catch (Exception ex)

{

System.out.print("B");

}

finally

{

System.out.print("C");

}

System.out.print("D");

}

public static void badMethod() {}

}

A. AC B. BC

C. ACD D. ABCD

8. What will be the output of the program?

public class X

{

public static void main(String [] args)

{

try

{

badMethod(); /\* Line 7 \*/

System.out.print("A");

}

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

{

System.out.print("B"); /\* Line 12 \*/

}

finally /\* Line 14 \*/

{

System.out.print("C"); /\* Line 16 \*/

}

System.out.print("D"); /\* Line 18 \*/

}

public static void badMethod()

{

throw new RuntimeException();

}

}

A. AB B. BC

C. ABC D. BCD

9. 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 ");

}

}

}

A. Nothing. The program will not compile because no exceptions are specified.

B. Nothing. The program will not compile because no catch clauses are specified.

C. Hello world.

D. Hello world Finally executing

10. 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");

}

}

}

A. Ex0 caught

B. exception caught

C. Compilation fails because of an error at line 2.

D. Compilation fails because of an error at line 9.

**Part-III (Who should do Part-III?)**

***If you are a person who loves to challenge yourself, train yourself till you tire and in short for those who aspire to become extra intellect, this is for you***

**Bank Application extension with Exception Handling**

Hint: You just need to extend the Exception class to create your own Exception class. These are considered to be checked exceptions. The following InsufficientFundsException class is a user-defined exception that extends the Exception class, making it a checked exception. An exception class is like any other class, containing useful fields and methods. Your withdraw() method has to throw InsufficientFundsException if there is no minimum balance or if amount requested > balance maintained in your account.