**EXCEPTION HANDLING SELF EXERCISE**

**A. Program.  
B. Objective Questions.**

**A. Program -**

Write a program which creates your own custom exception classes-

1. QaIncorrectChoiceException.
2. QaNumberValidationException.
3. QaTextValidationException.

Program takes two inputs in form of var args.

First input is choice. example - N (Number) or T (Text)

Second input is value. example - 32 , Luv , 50 , aaa.

If choice is other then N or T, then throw QaIncorrectChoiceException.

If choice is N and value is Text , then throw QANumberValidationException

If choice is N and value is Number, then check whether number is perfect square or not.

If choice is T and value is Number, then throw QaTextValidationException.

If choice is T and value is Text, then check whether text contains any vowel or not.

In case of any of the above exception in the code, also print “We are happy to help you, please contact us”.

**EXAMPLE -**

**CASE 1)** When choice is N and value is 16 , OUTPUT will be- Number is a perfect square.

**CASE 2)** When choice is N and value is 10 , OUTPUT will be- Number is not a perfect square.

**CASE 3)** When choice is N and value is hello , OUTPUT will be- QANumberValidationException.

**CASE 4)** When choice is T and value is hello , OUTPUT will be- Text contain vowels.

**CASE 5)** When choice is T and value is etc , OUTPUT will be- Text does not have vowels.

**CASE 6)** When choice is T and value is 5 , OUTPUT will be- QaTextValidationException.

**CASE 7)** When choice is Y and value is 16 , OUTPUT will be- QaIncorrectChoiceException.

**B. Objective Questions-**

**Q1.** What is the result of compiling and running the following code?

* public class Tester {
* public static void main(String[] args) {
* System.out.print("1");
* try {
* return;
* } catch (Exception e) {
* System.out.print("2");
* } finally {
* System.out.print("3");
* }
* System.out.print("4");
* }
* }

A) 1234

B) 13

C) 1

D) Compilation error

**Response: B : 13**

**Q2.** What is the expected output of compiling and running the following code?

* import java.io.IOException;
* class AirPlane {
* public AirPlane() throws IOException, RuntimeException {
* System.out.println("AirPlane");
* }
* }
* class AirJet extends AirPlane { } // line 7
* public class Tester {
* public static void main(String args[]) throws IOException { //line10
* new AirJet(); // line 11
* }
* }

A) AirPlane

B) Compile error at line 7, AirJet must declare a constructor that throws IOException or any of its supertypes

C) Compile error at line 10, main() must throw also RuntimeException

D) Compile error at line 11 ,new AirJet() must be within try/catch block

**Response: B) Compile error at line 7, AirJet must declare a constructor that throws IOException or any of its supertypes**

**Q3.** What is the expected output of compiling and running this code?

* class Father {
* public Father() throws RuntimeException {
* System.out.print("Father");
* throw new RuntimeException();
* }
* }
* class Son extends Father {
* public Son() throws RuntimeException {
* System.out.print("Son");
* }
* }
* public class Tester {
* public static void main(String[] args) {
* new Son(); // line 17
* }
* }

1. Compile error, an import to java.lang.RuntimeException is required
2. Father will be printed then a RuntimeException would be thrown
3. Compile error at line 17, new Son() should be within a try/catch block
4. SonFather
5. FatherSon
6. Son

**Response: B: Father will be printed then a RuntimeException would be thrown**

**Q4.** What is the result of compiling and running the following code?

* public class Tester {
* static void method(){
* throw new Exception();
* }
* public static void main(String[] args) {
* try {
* method();
* } catch (Throwable e) {
* try {
* throw new Exception() ;
* } catch (Exception ex) {
* System.out.print("exception");
* } finally {
* System.out.print("finally");
* }
* }
* }
* }

1. exception
2. finally
3. Exceptionfinally
4. Compilation error

**Response: D: Compilation error**

**Q5.** What is the expected output?

* import java.io.IOException;
* class Father {
* public Father() throws IOException {
* System.out.print("Father");
* throw new IOException();
* }
* }
* class Son extends Father {
* public Son() throws IOException {
* System.out.print("Son");
* }
* }
* public class Tester {
* public static void main(String[] args) {
* try {
* new Son();
* } catch (IOException e) {
* System.out.print("Inside catch");
* }
* }
* }

1. Compile error, constructors can't throw Exceptions
2. FatherSon
3. FatherInside catch
4. Father, then an IOException is thrown
5. Son
6. SonInside catch

**Response: C: FatherInside catch**

**Q6.** What is the expected result of compiling and running the following code?

* import java.io.IOException;
* class AirPlane {
* public AirPlane() {
* System.out.print("AirPlane");
* }
* }
* class AirJet extends AirPlane {
* public AirJet() throws IOException {
* try {
* throw new IOException();
* } catch (IOException e) {
* System.out.print("IOException is thrown in AirJet");
* }
* }
* }
* public class Tester {
* public static void main(String args[]) {
* try {
* new AirJet();
* } catch (IOException e) {
* System.out.print("IOException is thrown in Tester");
* }
* }

}

1. “AirPlaneIOException is thrown in AirJetIOException is thrown in Tester” will be printed
2. “AirPlaneIOException is thrown in AirJet” will be printed
3. “AirPlaneIOException is thrown in Tester” will be printed
4. Compilation error

**Response: B) “AirPlaneIOException is thrown in AirJet” will be printed**

**Q7.** When java NoClassDefFoundError is thrown?

1. When try to compile a non found java file
2. When try to run a non found java .class file
3. When try to compile a non found java file or try to run a non found java .class file

**Response: B**

**Q8.** What is the expected output?

* interface Foldable {
* public void fold() throws Exception ;
* }
* class Paper implements Foldable {
* public void fold() { // line 6
* System.out.print("Fold");
* }
* }
* public class Tester {
* public static void main(String args []) {
* Foldable obj1 = new Paper();
* obj1.fold(); // line 8
* Paper obj2 = new Paper(); // line 10
* obj2.fold();
* }
* }

1. FoldFold
2. Compilation error at line 6, fold() must declare at header throws Exception
3. Compilation error at line 8, unhandled exception

D) RuntimeException at line 8

E) Compilation error at line 10, unhandled exception

F) RuntimeException at line 10

**Response: Compilation error at line 8, unhandled exception**

**Q9.** What is the result of compiling and running the following code?

* public class Tester {
* public static void main(String[] args) {
* String stmt = "javachamp 2009";
* String[] arr = stmt.split(" ");
* try {
* int x = Integer.parseInt(arr[0]);
* System.out.print(x);
* } catch (Exception e) {
* System.out.print("catch");
* } finally {
* System.out.print("finally");
* }
* }
* }

1. 2009
2. 2009finally
3. **catchfinally**
4. No output will be produced
5. Compilation error
6. 2009catchfinally
7. javachampcatchfinally

**Q10.** What is the result of compiling and running the following code?

* public class Tester {
* public static void main(String[] args) {
* System.out.print("1");
* try {
* System.out.print("2");
* System.exit(0);
* } finally {
* System.out.print("3");
* }
* }
* }

1. 123
2. **12**
3. Compilation error , there should be at least one catch before finally

**Q11.** What is the result of compiling and running the following code?

* public class Tester {
* static void method() throws Exception {
* throw new Exception();
* }
* public static void main(String[] args) {
* try {
* method();
* } catch (Throwable e) {
* try {
* throw new Exception() ;
* } catch (Exception ex) {
* System.out.print("exception");
* } finally {
* System.out.print("finally");
* }
* }
* }
* }

1. “exception” is printed
2. “finally” is printed
3. “**exceptionfinally**” is printed
4. Compilation error

**Q12.** What is the result of compiling and running the following code?

* public class Tester {
* public static void main(String[] args) {
* String stmt = "javachamp 2009";
* String[] arr = stmt.split(" ");
* try {
* int x = Integer.parseInt(arr[1]);
* System.out.print(x);
* } finally {
* System.out.print("finally");
* }
* }
* }

1. 2009
2. finally
3. **2009finally**
4. No output will be produced
5. Compilation error

**Q13.** What is the output of the following code

* public class Tester {
* public static void main(String[] args) {
* Double d = -4.0;
* try {
* d /= 0;
* } catch (ArithmeticException e) {
* System.out.println("EXCEPTION!");
* } finally {
* System.out.println(d);
* }
* }
* }

1. NotANumber
2. NaN
3. EXCEPTION! -4.0
4. -**Infinity**
5. EXCEPTION! 0.0
6. Compilation fails

Q14. What is the result of compiling and running the following program?

* public class Tester {
* public static void main(String[] args) {
* try {
* throw new RuntimeException();
* } catch (RuntimeException e) {
* System.out.println("RuntimeException");
* } catch (ArithmeticException e) {
* System.out.println("ArithmeticException");
* } catch (Exception e) {
* System.out.println("Exception");
* }
* }
* }

1. RuntimeException is printed
2. Exception is printed
3. ArithmeticException is printed
4. **Compilation error**

**THANKS**