1. Which of the following are true about Error and Exception classes? (A)

* 1. Both classes extend Throwable.
  2. The Error class is final and the Exception class is not.
  3. The Exception class is final and the Error class is not.
  4. Both classes implement Throwable.

1. Given the code snippet below, try{

String obj=(String)new Object();

}

Which of these could be used in an appropriate catch block? (D)

* 1. NumberFormatException
  2. NullPointerException
  3. IllegalArgumentException
  4. ClassCastException

1. Analyze the code snippet below: (C)

class One{

public void showMessage()throws IOException{ }

}

class Two extends One{

@Override

public void showMessage() throws \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_{

}

}

Fill in the blank, to make the above program compile successfully

* 1. Exception
  2. Throwable
  3. RuntimeException
  4. FileNotFoundException

1. Analyze the code snippet below: (B)

public class Example {

public static void main(String[] args) { try{

int a=10/0;

System.out.println(“a=”+a);

}catch(Exception e){

System.out.println("Exception -1");

}catch(ArithmeticException e){

System.out.println("ArithmeticException");

}

}

}

What is the output?

1. Exception – 1
2. ArithmeticException
3. A= 10
4. Compilation Error

5. Analyze the code snippet below: (C)

public class Example {

public static void main(String[] args) {

try{

int a=10/0; System.out.println(“a=”+a);

}catch(NumberFormatException e){

System.out.println("Exception -1");

}finally{

System.out.println("Finally ");

}

}

}

What is the output?

1. Exception – 1 and Finally
2. 10 and Finally
3. Exception message and Finally
4. Compilation Error

6. Analyze the code snippet below: (C)

package com.core.ct;

public class Example {

public static void main(String[] args) {

try{

int a=10/0;

System.out.println("a ="+a); }catch(NumberFormatException e){

System.out.println("Exception -1");

}catch (ArithmeticException e) {

System.out.println("ArithmeticException");

}

finally{

System.out.println("Finally ");

}

}

}

What is the output?

1. Exception – 1 and Finally
2. 10 Exception -1 Finally
3. ArithmeticException and Finally
4. Compilation Error

7. Analyze the code snippet below: (C)

package com.core.ct;

public class Example {

public static void main(String[] args) { try{

String name="Java";

System.out.println(name.substring(1,5));

}catch(Exception e){

System.out.println("Exception in :"+name);

}

}

}

What is the output?

1. ava
2. java
3. Exception in : Java
4. Compilation Error

1. Which of the following events would most likely throw an exception at compilation time? (C)

* 1. Dividing any number by 0
  2. Attempt to search an array beyond the index position
  3. Trying to open a file
  4. Attempt to add two integer objects by + operator

1. Analyze the code snippet below: (C)

package com.core.ct;

public class Example {

public static void main(String[] args) {

System.out.println("Main method");

try{

doStuff();

}catch(RuntimeException e){

System.out.println("Exception - 1");

}finally{

System.out.println("Finally");

}

System.out.println("End of main");

}

public static int doStuff(){ throw new ArithmeticException();

}

}

What is the output?

1. Main method

Exception - 1

1. Main method Exception – 1

End of main

1. Main method

Exception – 1

Finally

End of main

1. Compilation Error

10. Analyze the code snippet below: (C)

package com.core.ct;

public class Example {

public static void main(String[] args) { System.out.println("Main method"); try{

int a=10/0; }finally{

System.out.println("Finally");

}

System.out.println("End of main");

}

}

What is the output?

1. Main method

Finally

1. Main method

Finally

End of main

1. Main method

Finally

Exception

1. Compilation Error

11. Analyze the code snippet below: (B)

package com.core.ct;

public class Example {

public static void main(String[] args) { System.out.println("Main method");

try{ return;

}finally{

System.out.println("Finally");

}

System.out.println("End of main");

}

}

What is the output?

1. Main method

Finally

1. Main method

Finally

End of main

1. Main method
2. Compilation Error

1. Exception and Error super class is \_\_\_\_\_\_\_\_\_\_ and its super class is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (C)
   1. RuntimeException , Object
   2. CheckedException, Object
   3. Throwable, Object
   4. Throwable, Exception

1. Any user-defined exception class is a subclass of the \_\_\_\_\_ class (D)
   1. NumberFormatException
   2. NullPointerException
   3. Exception/RunTimeException
   4. None of the above

1. By using throws clause, how many exceptions can be thrown in method declaration? (D)
   1. 1 B. 2
   2. 3
   3. Any number of exceptions

1. Which are the keywords used in java exception handling (C)
   1. try, catch, finally, thrown
   2. try, catch, finally, throws
   3. try, catch, finally, throws, throw
   4. none of the above

1. Which of the following are the valid code snippets to override the **show()** method in the **Example2** class**.** class One extends Exception{ (B)

}

class Two extends One{

}

class Three extends Two{

}

class Example1{

public void show()throws Three{ }

}

class Example2 extends Example1{

//show method overriding

}

1. public void show()throws IOException{ }
2. public void show(){ }
3. public void show() throws Two
4. public void show() throws One
5. public void show() throws Three

17. What is the output of the following code snippet? (B)

package spaneos.demo; class Example1 {

public int getResult(int a, int b) {

System.out.print("I "); return a / b;

}

}

class Example2 extends Example1 {

public int getResult(int a, int b) {

int res = 0;

try {

res = super.getResult(a, b); System.out.println(" am ");

} catch (Exception e) {

System.out.print(" am an ");

throw e;

}

return res;

} }

public class Example { public static void main(String... args) {

Example2 obj = new Example2();

try {

int res = obj.getResult(10, 0); System.out.println("The result is " + res);

} catch (Exception e) {

System.out.print(" Exception");

}

}

}

1. I am Exception
2. I am an Exception
3. I an am Exception
4. am I Exception
5. None of the above

18. Which of the following modification makes your program compilation and execution successful? (C)

package spaneos.demo;

class InvalidAgeException extends Exception{

//Code goes here

}

class Employee { private int age; public void setAge(int age) {

if(age<=0)

throw new InvalidAgeException(); this.age=age;

}

}

public class Example { public static void main(String... args){

Employee obj=new Employee(); try{

obj.setAge(-10);

}catch(InvalidAgeException e){

System.out.println("Invalid age exception");

} } }

1. Main method with throws InvalidAgeException
2. setAge() with the throws InvalidAgeException
3. Making InvalidAgeException as runtime exception
4. Remove the throw class in setAge()
5. None of the above

19. What is the output of the following code snippet? (C)

public class Example {

public static void main(String... args) { int a = 100, b = 0, c; try {

System.out.print("You"); if (b == 0)

throw new Throwable("sorry!"); c = a / b;

System.out.println("Result is :" + c);

} catch (Exception e) {

System.out.println(" are into a simple problem " + e.getMessage());

} catch (Throwable e) {

System.out.println(" are into a big problem " + e.getMessage());

}

}

}

1. You are into a big problem sorry!
2. You are into a simple problem sorry!
3. Result is : 0 and exception message
4. No output but it displays exception message
5. You followed by exception message

20. What is the output of the following code snippet? (A)

public class Example {

public static void main(String... args) { int a = 100, b = 0, c;

try {

System.out.print("You"); if (b == 0)

throw new Throwable("Sorry!"); c = a / b;

System.out.println("Result is :" + c);

} catch (Exception e) {

System.out.println(" are into a simple problem " + e.getMessage());

} catch (Throwable e) {

System.out.println(" are into a big problem " + e.getMessage());

}finally{

System.out.print("Don't worry");

}

System.out.print(" We help you to solve your prob");

}

}

A. You are into a simple problem Sorry!

Don't worry We help you to solve your prob B. You are into a simple problem Sorry! Don't worry

C. You are into a simple problem Sorry! We help you to solve your prob D. You are into a big problem Sorry!

Don't worry We help you to solve your prob E. You are into a simple problem Sorry!

Followed by exception message