Q1) What is Exception?

It is a type of event that can be handled and the other statements of the application can be

executed successfully.

Or

An exception is an object that represents an error or other unusual condition that has occurred during the execution of a program.

Q2) Write the usage of following?

a. try block

* Use the try block to enclose the code that might throw an exception.

* It must be followed by either a catch block or a finally block.

b. catch block

* Use the catch block to handle the exception thrown in the try block.

* It must be used after a try block or another catch block.

c. finally block

* Use the finally block to execute code regardless of whether an exception is thrown or not.

* It is typically used to release resources, close connections, etc.

d. throw keyword

* Use the throw keyword to explicitly throw an exception from a method.

* It is used to throw a custom exception or a runtime exception.

e. throws keyword

* Use the throws keyword to declare that a method may throw an exception.

* It is used to declare checked exceptions that a method might throw.

Q3) Can I define multiple catch blocks with one try block?

Yes, you can define multiple catch blocks with one try block

Q4) Can I define multiple finally block with one try block?

No, you cannot define multiple finally blocks with one try block

Q5) Can I define any other statements between try & catch, try & finally,catch and finally?

No, you cannot define any other statements between try & catch, try & finally,catch and finally

Q6) Can I define finally block after try block but before catch block?

No, you cannot define a finally block after the try block but before a catch block.

Q7) Can I define try block without catch block?

Yes, you can define a try block without a catch block.

Q8) Can I define try block without catch block and finally block?

No, you cannot define a try block without either a catch block or a finally block.

Q9) Can I define multiple try-catch blocks in one method?

Yes, you can define multiple try-catch blocks in one method.

Q10) Can I define throw statement immediately after return statement ?

No, you cannot define a throw statement immediately after a return statement.

Q11) Can I define return statement immediately after throw statement ?

No, you cannot define a return statement immediately after a throw statement.

Q12) What is difference between checked and unchecked exception?

Checked Exceptions

These are also called as Compile time exceptions.

- Extend the Exception class (except for RuntimeException)
  - Must be handled (caught or declared) at compile time
  - Examples: IOException, SQLException, FileNotFoundException

Unchecked Exceptions

These are also called as Runtime time exceptions.

- Extend the RuntimeException class
  - Need not be handled (caught or declared) at compile time
  - Examples: NullPointerException, ArrayIndexOutOfBoundsException, ClassCastException

Q13) What is difference between throw and throws keyword?

throw keyword

- throw is a keyword.
- It is used to throw the exceptions explicitly.
- You can throw any checked or unchecked exceptions.
- You can throw any Built-in or user-defined exceptions.

Syntax

throw <throwableTypeObjectRef>

throws keyword

- throws is a keyword.
- throws is used to propagate the exceptions to the caller method by specifying at method level.
- You can define any checked or unchecked exceptions at method level.
- You can define any Built-in or user defined exceptions at method level.

Q14) What is custom exception? Write the steps to define custom exception.

A custom exception is a user-defined exception class that extends the Exception class.

Steps:

1. Extend the Exception class.

2. Choose a meaningful name for your exception class.

3. Add a constructor that takes a String argument (error message).

4. Add methods (optional).

5. Use the custom exception with the throw keyword.

Q15) Can I handle the error?

Yes, you can handle the error.

Q16) What is the difference between exception and error?

a) Exception is a type of problem that can be handled and you can continue your program execution.

b) Exception occurs because of programming mistake.

c) Error is a type of problem that should not be handled and your program should be terminated.

d) Error occurs due to lack of resources.

e) Exception and Error are always occurs at runtime only.

f) Error coming at compile time is called as syntectical error and it is entirly different from error coming at runtime.

Q17) What is the reason of followoing excepiton and errors:

a. NoSuchMethodError:main

(if main() method is not found upto jdk1.6.)

b. NoClassDefFoundError

(if class name is wrong while executing.)

c. StackOverflowError

In recursive method call, if memory is not available in stack.)

d. OutOfMemoryError

(While creating the object, if there is no memory in heap.)

int arr[]=new int[54453453];

e. UnsupportedClassVersionError

(If JVM version is lower then compiler version.)

f. ExceptionInInitializerError

(if exception occurs at static variable or static block while loading the class.)

static int p;

static {p=Integer.parseInt("SRI");}

}

g. ClassCastException

(If you try to convert superclass object into subclass object whitout having the ref. of

subclass.)

Object ob1 = new Object();

String st1=(String)ob1;

h. IllegalStateException

Thread t= new Thread();

t.start();

t.start();//Here this err will occur

i. IllegalArgumentException

t.setPriority(30);

Thread priority must be b/w 1-10.

Q18) What are the new features in exception handling added from Java 7?

1. Multi-catch Block
2. try-with-resources

Q19) Explain Catching Multiple Exceptions using Single catch block.

・It is new features from Java 7.

・Using this, single catch block can handle more than one type of exceptions.

・It can reduce code duplication.

you can catch multiple exceptions in a single catch block using the "|" operator, like this:

catch (ExceptionType1 | ExceptionType2 | ExceptionType3 e)

Q20) What is the use of try with Resource?

With try-with resources try block can be used without catch and finally blocks also.

or

- Automatically close resources (like files, connections, sockets) after use

- Ensure resources are closed even if an exception occurs

- Reduce boilerplate code and improve code readability