Exception,imports and modifiers

Assignment Solutions

PW SKILLS

1.What is an exception in Java?

Ans: An exception is an abnormal event that disrupts the flow of normal program execution that unless handled could lead to the termination of that program. In real-world scenarios, a program could run into an exception if it tries to access a corrupted/non-existent file or if a network error happens or if the JVM runs out of memory, or if the code is trying to access an index of an array which does not exist and so on.

2. What are the types of Exception?

Ans: Exceptions can be categorised into two ways:

1. Built-in Exceptions
   - Checked Exception
   - Unchecked Exception
2. User-Defined Exceptions

3. How are exceptions handled in Java?

Ans: In Java, exceptions could be handled in the following ways:

try-catch block: The try section holds the code that needs to be normally executed and it monitors for any possible exception that could occur. The catch block "catches" the exception thrown by the try block. It could consist of logic to handle failure scenario or the catch block could simply rethrow the exception by using the "throw" keyword.

finally block: Regardless of whether an exception has occurred or not, if we need to execute any logic, then we place it in the final block that is usually associated with the try-catch block or just with the try block. The final block is not executed when System.exit(0) is present in either the try or catch block.

4. What are the important methods defined in Java's Exception Class?

Ans: The throwable class is the base class of Exception. Exception and its subclasses do not provide specific methods. All the methods that could be used by Exception are defined in the Throwable class. Some of the most important methods are as follows:

String getMessage(): This method returns the Throwable message of type String. The message could be given at the time of creating an exception by using the constructor.

String getLocalizedMessage(): This method can be overridden by the Exception and its subclasses to give locale-specific messages to the calling program. The implementation of this method in the Throwable class by default uses the getMessage() that returns the exception message. Hence, to use this method, it should be overridden to avoid this default behaviour.

synchronized Throwable getCause(): This returns the exception cause if it is known. If the cause is not known, it returns null. The cause is returned if it was provided via the constructors requiring Throwable or if it was set after the creation of exception by using the initCause(Throwable) method. Exception and its subclasses may override this method to return a cause set by different means.

String toString(): This returns the Throwable information in String format consisting of the Throwable class name and localized message.
void printStackTrace(): As the name indicates, this prints the stack trace data to the standard error stream. By default, it prints on the console. But, there are overloaded versions of this method where we can pass either PrintWriter or PrintStream as an argument to write stack trace data to a specific file or stream

5.What are runtime exceptions in Java?
Ans: Runtime exceptions are those exceptions that occur at the run time of the program execution. These exceptions are not noticed by the compiler at the compile time and hence the program successfully gets compiled. Therefore, they are also called unchecked exceptions. All subclasses of the java.lang.RunTimeException class and java.lang.Error class belongs to runtime exceptions. Examples of runtime exceptions include NullPointerException, NumberFormatException, ArrayIndexOutOfBoundException, StackOverflowError, ClassCastException, ArithmeticException, ConcurrentModificationException, etc

6. What is the difference between the throw and throws keywords in Java?
Ans: The throw keyword allows a programmer to throw an exception object to interrupt normal program flow. The exception object is handed over to the runtime to handle it. For example, if we want to signify the status of a task is outdated, we can create an OutdatedTaskException that extends the Exception class and we can throw this exception object as shown below:

```
if (task.getStatus().equals("outdated")) {
    throw new OutdatedTaskException("Task is outdated");
}
```

The throws keyword in Java is used along with the method signature to specify exceptions that the method could throw during execution. For example, a method could throw NullPointerException or FileNotFoundException and we can specify that in the method signature as shown below:

```
public void someMethod() throws NullPointerException, FileNotFoundException {
    // do something
}
```

7. What is the difference between error and exception in Java?
Ans: Errors are mainly caused by the environment in which an application is running. For example, OutOfMemoryError happens when JVM runs out of memory. Where as exceptions are mainly caused by the application itself. For example, NullPointerException occurs when an application tries to access null object.

8. Explain the hierarchy of Exceptions in Java?
Ans: In the java exception class hierarchy, the class at the top is the Throwable class, which is a direct subclass of the Object class. Throwable has two direct subclasses Exception and Error, which has again respective sub-classes The diagram below shows the standard exception and error classes defined in Java, organized in the Java exceptions hierarchy:

Error: Error caused by the lack of system resources such as heap memory is not available etc. Errors are non-recoverable. During the runtime of a program if any error occurs we will not be able to handle it.

9. Define Packages in Java.
Ans: A 'package' can be defined as a group of related types (classes, interfaces, enumerations and annotations) providing access protection and name space management.

10. What are Access modifiers in java?
Ans: Access Modifiers are those modifiers that are used to restrict the visibility of classes, fields, methods, and constructors.
Java supports four types of access modifiers:
Private
Default

Protected
Public

a) Private: Private members of a class can be accessed only within the class. It cannot be accessed from outside the class.

b) Default: Default members of a class are accessible within the same package due to visible only within the package. They cannot be accessed from outside the package.

c) Protected: Protected members of a class are visible within the package. Therefore, we can only access within the package but can be accessed to the subclasses outside the package through the inheritance only.

d) Public: Public members are visible anywhere. So, we can access it anywhere within or outside the package.

11. What are non-access modifiers in Java?
Ans: There are four non-access modifiers in Java. They are as follows:

Static
Final
Abstract
Synchronized

a) Static: This modifier is used to check that a member is a class member or instance member. If you declare a class as static, this class will be executed first.

b) Final: Final is a keyword that is used to restrict the users. In other words, it is used to restrict further modification of a class, field, or method. If a class is declared as 'final', the class cannot be subclassed.

c) Abstract: Abstract is a keyword that is used with a class or a method. An abstract class or abstract method is used for further modification. If a class is declared as 'abstract', the class cannot be instantiated.

d) Synchronized: It is used to achieve thread safeness. Only one thread can enter in a synchronized method or block at a given time.