Java

Assignment No- 28

Q.1 Can an abstract method have static qualifier?

→ No, an abstract method cannot have the static qualifier in Java.

Why?

- Abstract methods require overriding.
- An abstract method is meant to be overriden by subclasses.
- A static method belongs to the class itself, not to instances and cannot be overridden.
- Static method do not support dynamic binding.
- Abstract methods rely on dynamic method dispatch.
- Static methods use compile-time binding meaning they cannot be overridden dynamically.

Q.2 What kind of thread is the garbage collector thread?

→ The garbage collector thread in java is daemon thread that runs in the background to automatically reclaim unused memory by removing unreferenced objects.

- Features -

Thread Type - Daemon Thread (runs in the background)
Priority - Lower than user threads
Invocation - Runs automatically but can be requested using System.gc()
LifeCycle - Managed by JVM, stops when no user threads remain
Purpose - Frees memory by removing unreachable obj

- Why is the Garbage Collector a Daemon Thread?

Daemon threads are low-priority background threads that perform system level tasks.

The GC thread does not block the ~~gre~~ program execution and runs only when needed.
- The JVM terminates daemon threads automatically when all user threads finish.

Q3] What is Daemon Thread?
→ A daemon thread in Java is a background thread that provide supportive tasks for user threads. It runs continuously in the background and terminates automatically when all non-daemon threads have finished execution.
- Purpose - Supports user threads
- priority - Lower than user threads
- Lifecycle - Ends when all user threads finish
- Created By - JVM or manually set by setDaemon
- Execution control - JVM may not guarantee full execution before termination.

Q4] What is use of finally block?
→ The finally block in java is used for code that must always execute, regardless of whether an exception occurs or not. It is mainly used for resource cleanup
- Uses
- Resource Cleanup - closes files, databases or network connections
- Ensuring Code Execution - Runs even if an exception is thrown
- Handling Exceptional Scenarios - Prevents memory/ resource leaks

Syntax -
try { // code that may throws an exception }
catch (Exception e) { // Handling the exception }
finally { // code that will always execute }

Q5] What is mutable and immutable object?
→ In java objects are classified as mutable or immutable based on whether their state can be changed after creation.

• Mutable Object
- Can be ~~eta~~ modified after creation.
- Methods exist to change the object's state
- Used when data needs frequent updates.

Immutable objects
- Cannot be modified after creation
- Any modification results in a new object being created
- Used for thread safety and caching (e.g. String, int)

Q6] What is basic difference between String and StringBuffer object.
→ String (Immutable)
- Immutable - Once string object is created, its value cannot be changed.
- Memory Usage - Since every modification creates a new object, it consume more memory.
- Performance - Slower when modifying strings frequently because it keeps creating new objects.
- Thread - safety - Inherently thread - safe because String objects cannot be modified.
- Stored in String Pool - Java stores string literals in

the string Pool to optimize memory

ex- String s = "Hello";
s = s.concat("world");
System.out.println(s);

A new object is created and s now points to the new object.

StringBuffer (Mutable)

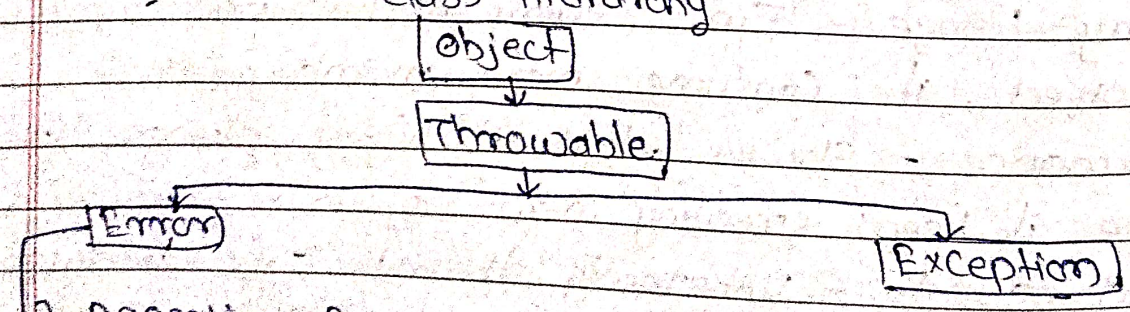
-Mutable - StringBuffer allows modification of the same object without creating a new one.
-Memory Efficient - Since it modifies existing objects it consumes less memory.
- Performance - Faster for operations like append(), insert(), delete().
- Thread - Safety : Synchronized making it safe for use in multi- threaded applications.

ex - StringBuffer sb = new StringBuffer("Hello");
sb.append("World");
System.out.println(sb);

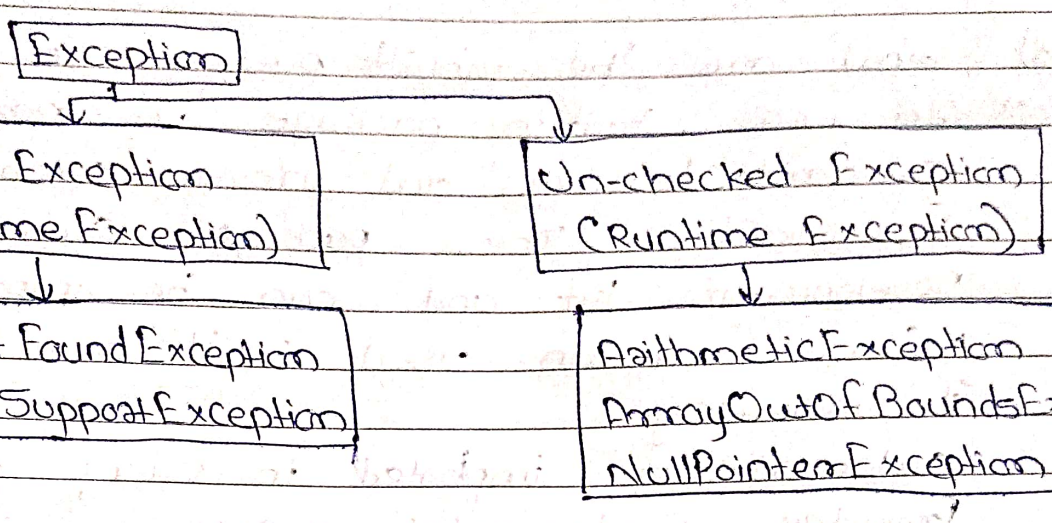
Q7] What is the base class for Error and Exception.

→ In java, Both Error and Exception are subclass of Throwable, which is the root class for all types of exception and errors.

- Class Hierarchy



1) Assertion Error
2) OutOfMemoryError



• Throwable (Base class)
- Superclass of all exceptions and errors in Java.
- Provides methods like printStackTrace(), getMessage(), and toString().

• Exception (Recoverable Errors)
- Used for normal application errors that can be handled using try-catch.
- Subclasses — Checked Exception (must be handled)
  IOException, SQLException
  — Unchecked Exception (extend RuntimeExcep)
  ~~Null~~RuntimeException
  NullPointerException,
  ArrayIndexOutOfBoundException,

• Error (Critical System Issues)
Represents serious problems that an application should not try to handle.
ex- outOfMemoryError
StackOverflowError
VirtualMachineError.

Q8] What are the inbuilt package in java

→ In java, built in package are part of the java standard Library and provide commonly used functionalities. These packages come with the java Development kit and can be improved as needed

• Some common used built in packages

1] Java.lang
automatically imported in every java program
Provides fundamental classes.

2] java.util
provides utility classes for data structure, data/tim collections, etc.

3] Java.io
Used for input and output operation.

4] java.nio
provide advanced file handling and I/O operations.

5] java.net
Sopports networking operations like HTTP, socket and URLs.

6] java.sql
Used for database connectivity

7] java.text
Used for text formatting and parsing

8] java.time
provides modern date and time API

9] javax.swing
Used for building GUI applications

10] javafx.*
Used for modern GUI development in Java

11] Java.security
Provides security related functionalities.

Q9] Is JVM a compiler or interpreter?

→ In java : Virtual Machine is both an interpreter and Just-In-Time compiler. but it is not a traditional Compiler like javac.

How it works:

1) Compilation Phase (javac - Compiler)

The Java compiler (javac) converts java source code into bytecode

Bytecode is an intermediate representation, not machine code.

2) Execution Phase (JVM - Interpreter + JIT Compiler)

The JVM interprets bytecode and converts it into machine code line by line at runtime.

However, modern JVMs use a Just In Time compiler to optimize performanced ~~to~~ by compiling frequently used bytecode into native machine code.

So JVM is

An interpreter (executes bytecode line by line initially)

A JIT compiler (compiles frequently used code into native machine code for better performance)

Q10] what is the use of final keyword.

→ The final keyword in java is used to restrict modification. It can be applied to variable, methods & classes with different effects.

1) final with Variables

Prevents changing the value after initialization

2) final with methods (prevent overriding)

Prevents a subclass from overridding a method.

3) final with classes (prevent inheritance)

Prevents a class from being extended (inherited

3) final with Reference Variables

A final reference variable cannot be reassigned but its internal state can change (if it is mutable)