# Assignment 03 Application Design: Patterns and Frameworks 44642

Answer **all** the questions below. In your answer for each question explain with sample code or image whichever is preferable.

1. What are generics?

* Generics enables developers to create flexible and reusable code that works with multiple data types, while also promoting code safety by checking types at compile time. A generic method in Java, for example, can handle any type of list, including those with different types of elements, saving time and avoiding repetition. Overall, generics are a powerful tool for making programming more efficient, adaptable, and less error prone.

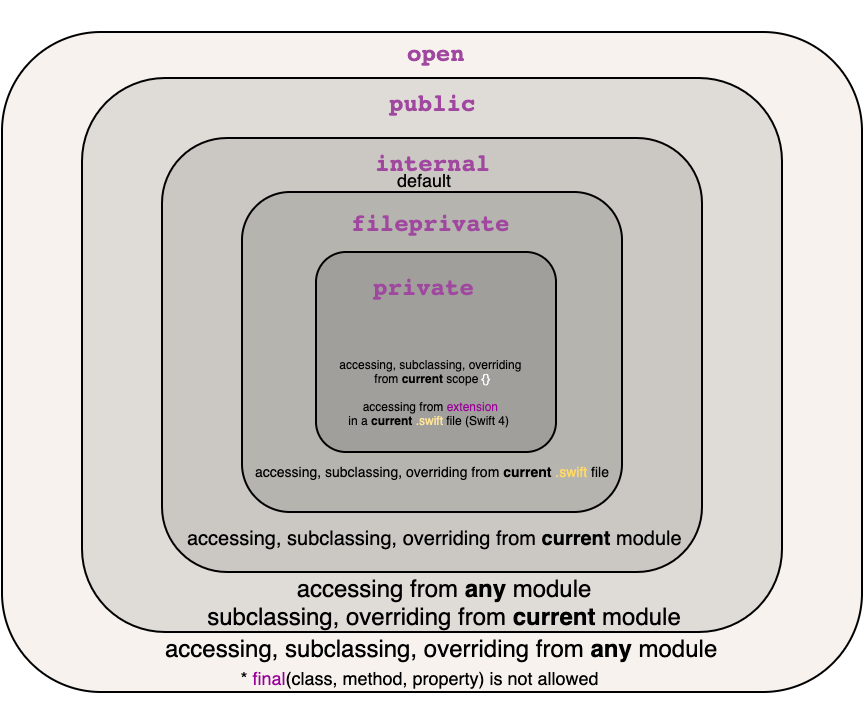
Graphical user interface, text, application

Description automatically generated

1. Can we change the scope of the overridden method in the subclass for private, public, default and protected? Explain how can it be changed for each scope?

* When a method is overridden in a subclass, its visibility (public, protected, default, private) can't be reduced compared to the superclass method.
* If the superclass method is private, it can't be overridden.
* A default method can be overridden in the same package subclass.
* A protected method can be overridden in the same package subclass or different package subclass.
* A public method can be overridden in any subclass.

To increase visibility, the subclass method can be declared with higher visibility.



1. What is the covariant return type?

Covariant return type is a Java feature that allows a subclass to have a return type that is a subclass of the return type of its superclass. This makes the code more flexible and readable, as it allows returning more specific types without casting. It was introduced in Java 5. For example, a subclass can return Dog or Cat while the superclass returns Animal. It simplifies code and improves efficiency.

Graphical user interface, text, application

Description automatically generated

1. Can we override the static and private methods? Why?

Java does not allow the overriding of static and private methods because they are associated with the class rather than a specific instance. These methods are not inherited by subclasses, so they cannot be overridden. Since private methods are not visible to subclasses and static methods are associated with the class rather than its instance, it is impossible to override them in Java.

Graphical user interface, text, application

Description automatically generated

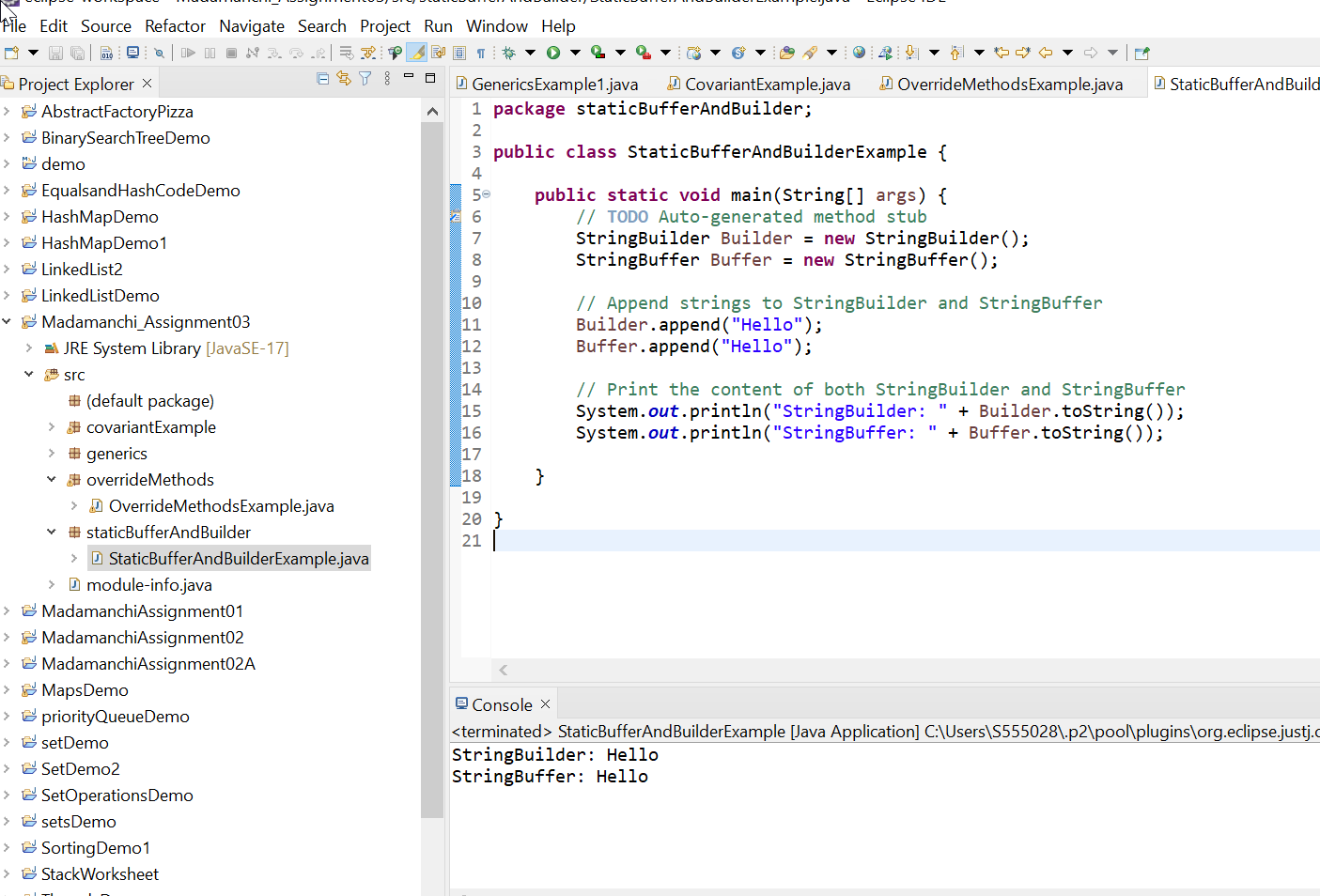
1. Difference between String Buffer and StringBuilder?

String Buffer: Used for String manipulation in java.

* 1. String Buffer is thread-safe, which means it can be used by multiple threads simultaneously without any issue. It is slower than StringBuilder due to its thread-safety.

StringBuilder: Used for String manipulation in java.

* 1. StringBuilder is not thread-safe, so it can only be used by one thread at a time. It is faster than String Buffer due to its lack of thread-safety.



1. Difference between String class and String Buffer?

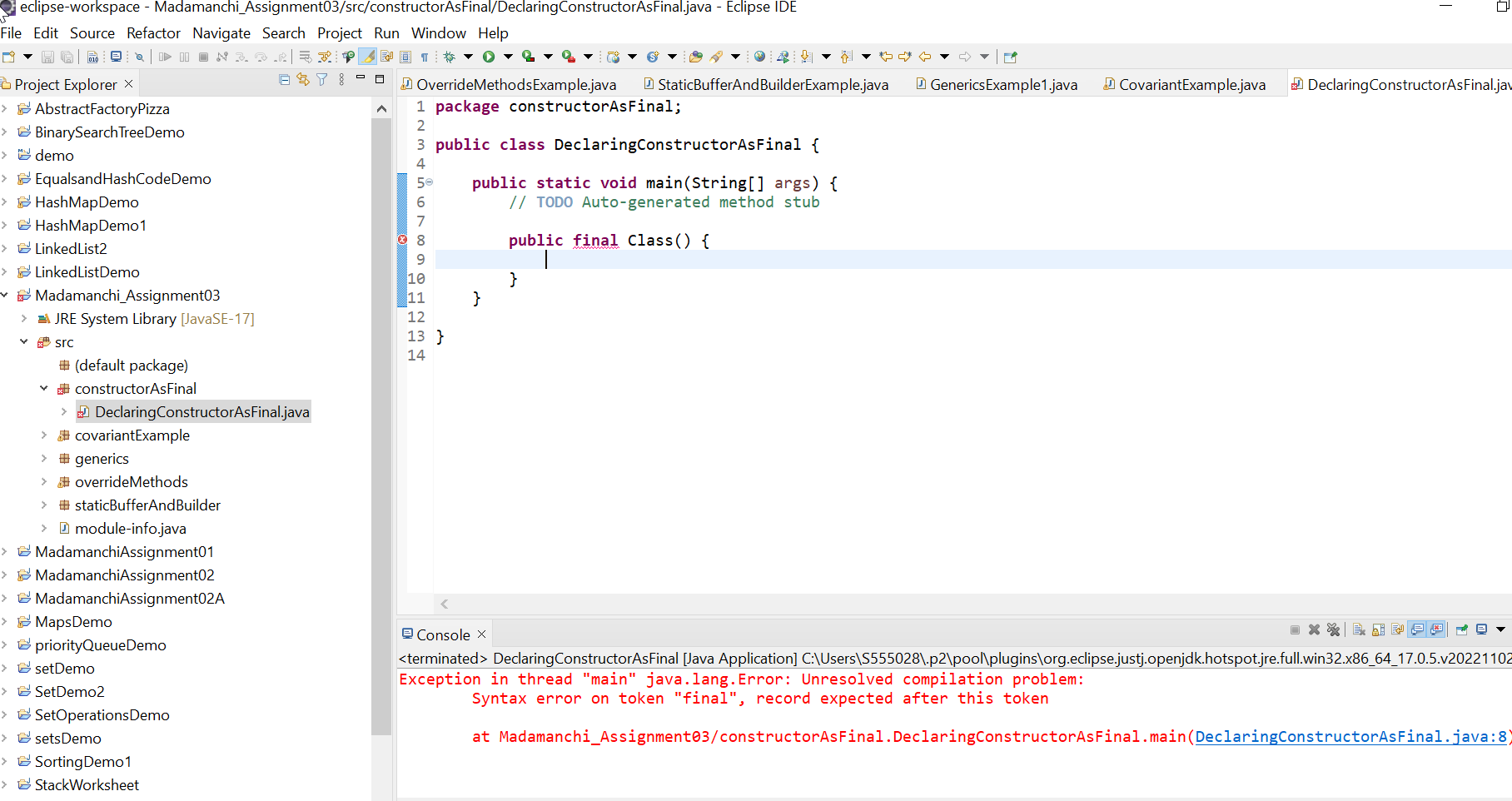
Strings are immutable, StringBuffer are mutable. The main difference between String class and StringBuffer in Java is that Strings are immutable, meaning their values cannot be modified, while String Buffers are mutable, meaning their values can be changed. When a string value is changed, a new string object is created in memory, whereas StringBuffer can change its value in the same object without creating a new one. This makes StringBuffer more efficient when working with large or frequently changing strings.

Graphical user interface, text, application

Description automatically generated

1. Can we declare constructor as final?

In Java, constructors cannot be declared as final. Attempting to do so will result in a compilation error. Here's an example:



1. Can we have try without catch block in java?

In Java, it is allowed to use a try block without a catch block, but it is mandatory to include a finally block. The finally block will execute irrespective of whether an exception is thrown or not in the try block. Below is an example of a try block without a catch block:

Graphical user interface, text, application

Description automatically generated

1. What is try with the resource?

In Java, the try-with-resources feature automatically closes resources such as files, network sockets, and database connections at the end of a block. This feature eliminates the need for a finally block and requires that the resource implements the Auto Closeable interface. A file can be opened and read using try-with-resources, and the file will be automatically closed at the end.

~~Graphical user interface, text, application

Description automatically generated~~

1. Can we modify the throws clause of the superclass method while overriding it in the subclass?

In Java, the try-with-resources feature automatically closes resources such as files, network sockets, and database connections at the end of a block. This feature eliminates the need for a finally block and requires that the resource implements the AutoCloseable interface. For example, a file can be opened and read using try-with-resources, and the file will be automatically closed at the end of the block.

Text, application

Description automatically generated

1. What is an association, aggregation, and composition in UML?

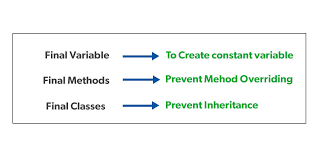
In UML, an association is a link between two or more classes that represents how objects of one class are related to those of another. Aggregation is a specific type of association where one class contains or has other classes or objects, creating a "has-a" relationship. Composition is another type of association where the contained object is an integral part of the container object, creating a "part-of" relationship.

Diagram

Description automatically generated

1. Difference between final, finally and finalize()?

The "final" keyword in Java is utilized to declare constants that cannot be modified after they are assigned a value. For instance, if you declare a variable as "final int x = 5", you cannot assign a different value to x later in your code. The "finally" block in Java exception handling is used to guarantee that a block of code executes regardless of what happens. It is usually employed for releasing resources such as file handlers, network sockets, and database connections. The finally block executes after the try block and any catch block(s) have executed. The "finalize()" method is a special method in Java that is called by the garbage collector when an object is no longer in use and is about to be destroyed. It allows objects to perform any final operations before they are deleted from memory. For instance, if an object has an open file or network connection, it can close it in its finalize() method. However, the usage of the finalize() method is not recommended in modern Java development because it can result in performance issues. To summarize, final is used for constant declaration, finally is employed for executing code that must run regardless of exceptions, and finalize() is a method that enables objects to perform final operations before they are destroyed.



1. Difference between Vector and ArrayList?

Both Vector and ArrayList provide dynamic arrays to store elements, but they have some key differences. Vector is synchronized and thread-safe, while ArrayList is not. Vector doubles its array size when it runs out of space, while ArrayList increases its size by 50% of the current size. Vector has extra methods that allow it to be used as a stack or a queue, while ArrayList only supports list operations. Finally, ArrayList is faster than Vector when it comes to increasing its size.



1. What are the different ways to make ArrayList methods synchronized?

- There are two ways to create a Synchronized ArrayList.

1. Collections.synchronizedList() method.

2. Using CopyOnWriteArrayList.

Graphical user interface, text, application

Description automatically generated

1. Difference between Hash table and Hash Map?

In Java, HashTable and HashMap are both used to store key-value pairs, but HashTable is an older class while HashMap is newer. The main difference is that HashTable is synchronized, making it safe to use in multi-threaded applications, while HashMap is not synchronized by default but can be made thread-safe. Additionally, HashTable doesn't allow null keys or values, while HashMap allows one null key and multiple null values.

Graphical user interface, text, application

Description automatically generated

1. In Java 8, explain how Hasp Map internally works?

HashMap in Java 8 stores key-value pairs in an array, with each element being a linked list of nodes, and the hash code of the key determines the index. It uses a hashing algorithm to distribute the keys, making it faster to search for a key. The load factor affects the performance and determines when the array needs to be resized. Java 8's implementation of HashMap uses tree-based buckets optimization to enhance performance when numerous elements are stored in the same linked list.

Diagram

Description automatically generated

1. Difference between fail fast and fail-safe iterator?

Fail-fast and fail-safe are two types of iterators in Java that are used to traverse collections. Fail-fast iterators throw a ConcurrentModificationException if the collection is modified during iteration, while fail-safe iterators do not throw an exception and create a copy of the collection before iterating. Fail-fast iterators quickly detect any changes made to the collection during iteration and prevent further operations to maintain consistency, while fail-safe iterators create a snapshot of the collection to avoid any inconsistency between the original collection and its copy during iteration.

Ex:

ArrayList<Integer> numbers = // ...

Iterator<Integer> iterator = numbers.iterator();

while (iterator.hasNext()) {

Integer number = iterator.next();

numbers.add(50);

}

Iterators on Collections from java.util.concurrent package such as ConcurrentHashMap, CopyOnWriteArrayList, etc. are Fail-Safe in nature.

Ex:

ConcurrentHashMap<String, Integer> map = new ConcurrentHashMap<>(); map.put("First", 10);

map.put("Second", 20);

map.put("Third", 30);

map.put("Fourth", 40);

Iterator<String> iterator = map.keySet().iterator();

while (iterator.hasNext()) {

String key = iterator.next();

map.put("Fifth", 50);

}

1. Can we start the thread twice?

In Java, a thread cannot be started twice. When an attempt is made to start a thread that has already completed its execution, it results in an "IllegalThreadStateException". Although a new thread with the same name can be created and started after the previous one has completed, starting the same thread again is not possible.

Graphical user interface, text, application

Description automatically generated

1. What are the different ways to create a thread in java? Which one is preferred?

In Java, creating a thread can be done either by implementing the Runnable interface or by extending the Thread class.

Implementing Runnable is preferred because it separates the concerns and enables better code reusability. It also allows for easier use of thread pools and concurrent task execution. However, extending the Thread class may be useful in certain situations, such as when there is a need to override some of the Thread class methods or to have more direct control over the thread's behavior. While implementing the Runnable interface, a class can still extend another class and create a thread by implementing Runnable.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. What are the different states a thread will go through?

A thread in Java can go through different states during its lifecycle. These states are: New: A thread is in this state when it is first created.

Runnable: A thread is in this state when it is ready to run, but is waiting for a processor to execute it.

Running: A thread is in this state when it is being executed.

Blocked: A thread is in this state when it is waiting for a monitor lock to be released. Waiting: A thread is in this state when it is waiting for another thread to perform a particular action.

Timed Waiting: A thread is in this state when it is waiting for a specified period of time. Terminated: A thread is in this state when it has completed its execution or has been terminated.

Diagram

Description automatically generated

1. What is Serialization? How do we achieve it?

Serialization in Java involves converting an object into a sequence of bytes for storage or transmission over a network. The Serializable interface must be implemented by the object's class to accomplish serialization. The Serializable interface doesn't have any methods, but it signals to the Java Virtual Machine (JVM) that the class can be serialized. During serialization, an object's instance variables are written to an output stream in a specific format, including information about the object's class, superclass, and the values of its instance variables.

Graphical user interface, application

Description automatically generated

1. What is immutable class? Is String class immutable?

Immutable classes in Java cannot be modified once they are created and have final instance variables and no mutator methods. The String class is immutable, and its value cannot be changed after creation. Modifying a String creates a new String object. This immutability makes String objects cacheable and thread safe.

Graphical user interface, text, application

Description automatically generated

1. Do immutable classes thread safe? If yes then how?

Yes, immutable classes are thread-safe as their state cannot be modified once created. This ensures that multiple threads can read the same object without needing synchronization. This makes immutable objects safe to use in concurrent environments.

Graphical user interface, text, application

Description automatically generated

1. Can we call the garbage collector explicitly? Will it trigger the garbage collector?

In Java, we can use the System.gc() method to explicitly request the garbage collector to run, but it is not guaranteed that it will run immediately. The JVM determines when to run the garbage collector and may delay or ignore the request.

Graphical user interface, text, application

Description automatically generated

1. What are Java 8 features? Explain all of them with examples?

Java 8 provides following features for Java Programming:

* Lambda expressions,
* Method references,
* Functional interfaces,
* Stream API,
* Default methods,
* Base64 Encode Decode,
* Static methods in interface,
* Optional class,
* Collectors class,
* ForEach() method,
* Nashorn JavaScript Engine,
* Parallel Array Sorting,

1. How to make a pure singleton?

Creating a pure singleton involves making sure that only one instance of a class exists during the application's lifetime. To do this, the class constructor is made private to prevent external instantiation, and a public static method is provided to access the single instance of the class. The method creates the instance only if it does not already exist; otherwise, it returns the existing instance. Additionally, the class is made final to prevent any subclass from creating another instance. This ensures that the singleton object is thread-safe and can be accessed from multiple parts of the code without creating multiple instances.

Text

Description automatically generated

1. How to make a singleton synchronized?

Synchronizing the getInstance() method is a way to make a singleton thread-safe. It ensures that only one thread can create an instance of the class, preventing multiple instances from being created. However, this can impact performance due to thread waiting times. Another option is to use double-checked locking, where synchronization only occurs if the instance is null. This approach improves performance by avoiding unnecessary synchronization.

Graphical user interface, text, application

Description automatically generated

Github Link: https://github.com/Harikamadamanchi/Madamanchi\_Assignment\_03.git

**Submission:**  Change the document name as ***Lastname*Assignment03** where *Lastname* is your Last name and submit.