Strings in Java

Assignment Questions

Assignment Questions

1. **What is a String in Java ?**

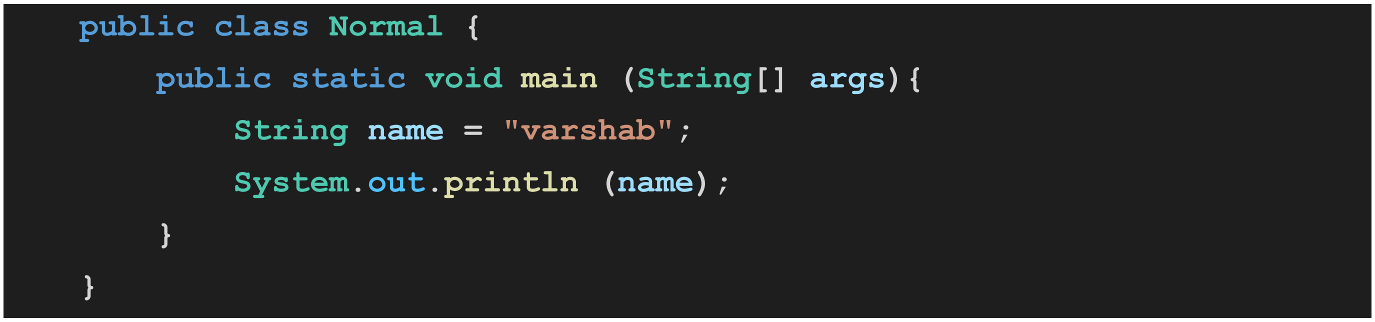
   Ans :

- **In Java, a String is a class that represents a sequence of characters. It is used to store and manipulate text-based data, such as words, sentences, or any other textual information.**

- **The String class is part of the Java Standard Library and provides many useful methods for working with strings.**

- **String objects are immutable, which means once a String is created, its value cannot be changed. If we want to modify a string, a new string object is created with the modified value.**

Ex :



```
public class Normal {
    public static void main (String[] args){
        String name = "varshab";
        System.out.println (name);
    }
}
```

Output :

varshab

2. Types of String in Java are ?

Ans :

- **Immutable String:**

a. An immutable string is a string whose value cannot be modified after it is created.

b. Once a String object is created, its value remains constant throughout its lifetime.

c. String immutability ensures safety and consistency, as strings can be freely shared among different parts of a program without fear of modification.

Ex :

```
public class ImmutableString{
    public static void main (String[] args){
        String name = "varshab";
        System.out.println (name);
        name.concat("chandel");
        System.out.println (name);
    }
}
```

Output :

```
varshab
varshab
```

- **In the above code the String chandel is not added in the string varshab because we use the string keyword which is immutable.**

3. In how many ways can you create string objects in Java ?

Ans :

- **Using String Literal :**

a. String literals are created by enclosing characters within double quotes.

b. When you create a string using a string literal, Java automatically checks the string pool for an existing string with the same value. If it exists, the existing string is reused; otherwise, a new string is added to the pool.

c. String literals are stored in the string pool, making them more memory-efficient, as duplicate strings are not created.

Ex :

String name = “varshab”;

- **Using the new Keyword and String Constructor :**

a. **You can use the new keyword along with the String constructor to create a new string object.**

b. **Unlike string literals, this method does not use the string pool, so a new object is created each time, even if a string with the same value already exists.**

Ex :

String name = new String (“varshab”);

- **Using StringBuilder or StringBuffer:**

a. **The StringBuilder and StringBuffer classes are used for mutable sequences of characters.**

b. **They provide methods to efficiently modify strings by appending, inserting, or deleting characters.**

c. **Once the desired changes are made, you can convert the StringBuilder or StringBuffer back to an immutable String using the toString() method.**

Ex :

StringBuilder name = new StringBuilder (“varshab”);

4. What is a string constant pool ?

Ans :

- **The String Constant Pool, often referred to as simply the "string pool," is a special area in the Java heap memory where the JVM stores a pool of unique string literals.**
- **It is a mechanism used by Java to conserve memory and improve performance for string operations.**
- **In the string pool, duplicate copies of the same string literal are not allowed. Instead, if multiple strings with the same value are encountered, the JVM reuses the existing string from the pool.**
- **This ensures that each unique string literal is represented by a single String object in memory.**
- **String pooling provides performance benefits and is especially useful when dealing with string literals in your Java code.**

5. What do you mean by mutable and immutable objects ?

Ans :

- **Mutable and immutable are terms used to describe the state and behavior of objects in programming languages like Java. They refer to whether the properties (fields) and content of an object can be changed after it is created.**

- **Immutable objects are useful in scenarios where you want to ensure that the object's state remains constant and cannot be accidentally modified by other parts of the code. They offer benefits like thread-safety, caching, and easier reasoning about code behavior.**

- **Mutable Objects :**

a. **A mutable object is an object whose state or content can be modified after it is created.**

b. **This means that you can change the values of its fields and properties, add or remove elements, or perform other modifications on the object's internal data.**

c. **In other words, the state of a mutable object is not fixed and can be altered throughout its lifetime.**

- **Immutable Objects:**

a. **An immutable object is an object whose state or content cannot be changed after it is created.**

b. **Once an immutable object is created, its values and internal data cannot be modified.**

c. **If you need to modify an immutable object, a new object with the modified values is created, leaving the original object unchanged.**

6. Where exactly is the string constant pool located in the memory ?

Ans :

- **In Java, the String Constant Pool is a part of the Java Runtime Environment's (JRE) heap memory. It is a special area within the heap where the JVM stores a pool of unique string literals created in the application.**

- **The heap memory is a region of memory used for the dynamic allocation of objects and arrays during the runtime of a Java application.**

- **It is separate from the stack memory, which is used for method call frames and local variables.**

- **The String Constant Pool is shared among all parts of the application, and it contains only one copy of each unique string literal. This is achieved by checking the pool before creating a new String object for a string literal.**

- **If a string with the same value already exists in the pool, the existing object is reused; otherwise, a new String object is added to the pool.**