1. List down Data Types in Java (with wrapper class name, memory size, range) in a tabular format.

|  |  |  |  |
| --- | --- | --- | --- |
| Data Type | Wrapper Class | Memory size | Range |
| byte | Byte | 1 byte | -128 to 127 |
| short | Short | 2bytes | -32768 to 32767 |
| int | Integer | 4 bytes | -2147483648 to 2147483647 |
| long | Long | 8bytes | -9223372036854775808 to 9223372036854775807 |
| float | Float | 4bytes |  |
| double | Double | 8bytes |  |
| char | Character | 2bytes |  |
| string | String | Depends on size |  |
| boolean | Boolean | 1bit | True or False |

2. Write the concepts discussed about strings in the class.

String: String is an object that represents sequence of char values. Objects of Strings are immutable which means a constant and cannot be changed once created.

Creating a string:

String s = “Madhuri”;

String s = new String(“Madhuri”); //using new keyword

Stack memory: It is used for static memory allocation and the execution of the string.

It contains primitive values that are specific to a method and references to objects referred from the method that are in a heap.

Variables inside the stack exist only as long as the methods that created them is running.

Must faster to allocate when compared to heap

Heap Memory: The Heap memory contains all objects are created, but Stack contains any reference to those objects. There is no size limit on heap

New objects are always created in Heap space and the reference to these objects are stored in stack memory.

These objects have global access and we can access them from anywhere in the application.

Slower to allocate when compared to stack.

3. Research and write what is String pool in java

String pool is a storage area in java heap where string literals stores.

Each time string literal is created, the JVM checks the string literal pool first. If the string already exists in the string pool, a reference to the pooled instance returns.

If the string does not exist in the pool, a new string object initializes and is placed in the pool.

String s1= ”Apple”;

JVM checks the same value exists in the string constant pool or not.

If yes, it occupies the already existing value.

If no, it creates a new string by itself and adds it to the string pool.

String s2= ”Mango”;

String s3= "Apple”;

String s4= new string(“Apple”);

The above statement creates the string in java heap. If we want to store the string literal in the sting pool, we should use the intern()method.

Here 1. s1==s2 the output will be false

2. s1==s3 the output will be true

3. s1==s4 the output will be false

The class is loaded when JKVM is invoked.

JVM looks for all the sting literals in the program.

1-First, it finds the variable s1 which refers to the literal “Apple” and it gets created in the memory.

A reference for the literal “Apple” is then placed in the string constant pool memory.

Then it finds another variable s2 which refers to the same string literal ”Mango”.

Then it finds another variable s3 which refers to the string literal “Apple”.

Now that JVM has already found a string literal “Apple”, both the variables s1 and s3 will refer to the same object.

s1 and s4 return false because s1 occupies in string pool and s4 occupies in java heap.