COMMENTS:-The Java Comments are the statements in a program that are not executed by the compiler and interpreted.

- Comments can be used to explain Java Code, and to make it more readable.
- It can also be used to prevent execution when testing alternative code.
    - Single-Line Comments
    - Multi-Line Comments

Single-Line Comments: Single-Line Comments start with two forward slashes(//).

- Any text between // and the end of the line is ignored by java(will not be executed).
- This example uses a single-line comment before a line of code.

Example:

```
Public class

{

  Public static void main(String[] args)

   {

   //This is a comment

     System.out.println(“Hello World”);

    }

 }
```

Multi-Line Comments: Multi-Line Comments starts with /* and ends with */.

- Any text between /* and */ will be ignored by java.
- This example uses a multi-line comments

Example:

```
Public class

{

  Public static void main(String[] args)

   {

   /* The code below will print the words Hello World to the screen , and it is is
amazing*/

     System.out.println(“Hello World”);

    }

 }
```

Collections:

- The Collections in Java provides an architecture to store and manipulate the group of objects,interfaces and classes.
- This java collection is a framework .This framework has several useful functions that have tons of useful functions,making a programmer task super easy.
- This framework provides many interfaces and classes
    - Interfaces are Queue,set,list,Deque.
    - Classes are priorityQueue,HashSet,ArrayList,Vector,LinkedList,LinkedHashSet.

Differences Between Index based for loop and enhanced for loop

Index Based for Loop	Enhanced for loop
1)Accessing the data through indexing	**1)**Data is traversed from start to end
2)In Index Loop we can increment reverse order/even/odd	**2)**We can also traverse in forward direction because it is not traversed in backword direction
3)Sometimes index may throw out of bound errors	**3)**The data can be access in direct

4) Syntax:
 	
for(initialization section ; conditional section ; increment/decrement section)
{
// Code to be executed
}
	4) Syntax:
	
	for(data-type variable : array | collection)
	{
	// Code to be executed
	}

5) Example:
 	
import java.io.*;
import java.util.*;
class GFG
{
public static void main(String[] args)
{
for (int i = 1; i <= 5; i++)
{
System.out.println("GFG!");
}
for (int i = 1; i <= 1; i++)
{
int x = 0;
}
}
}
	5) Example:
	
	import java.io.*;
	import java.util.*;
	classGFG
	{
	public static void main(String[] args)
	{
	int[] array = { 1, 2, 3, 4, 5, 6 };
	
	for (int a : array)
	{
	System.out.println(a);
	
	}
	}
	}

Package: Package is a way of organizing our files it uses directories for doing such activities.

- Package is a way of organizing our files it uses directories for doing such activities.
- A java package is a group of similar types of classes, interfaces and sub-packages.
- Package in java can be categorized in two form, built-in package and user-defined package.
- There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantages of package:

- Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- Java package provides access protection.
- Java package removes naming collision.

EXAMPLE:

```
package mypack;
public class Simple
{
 public static void main(String args[])
{
System.out.println("Welcome to package");
}
}
```

Java List:

- List in Java provides the facility to maintain the ordered collection.
- It contains the index-based methods to insert, update, delete and search the elements.
- It can have the duplicate elements also. We can also store the null elements in the list.

Example:

```
import java.util.*;

public class List

{

public static void main(String args[])

{

   List<String> list=new ArrayList<String>();

  list.add("Mango");

 list.add("Apple");

 list.add("Banana");
 list.add("Grapes");
 for(String fruit:list)
  System.out.println(fruit);
  }
}
```

JAVA ARRAYS:

- An array is a collection of similar type of elements which has contiguous memory location.
- It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.
- It is storing the data in the form of indexes.

Advantages:

- Code Optimization: It makes the code optimized, we can retrieve or sort the data efficiently.
- Random access: We can get any data located at an index position.

Disadvantages:

Size Limit: We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.

EXAMPLE:

```
class Array
{
public static void main(String args[])
{
int a[]=new int[5];
a[0]=10;
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;
for(int i=0;i<a.length;i++)
{
System.out.println(a[i]);
}
}
```

Wrapper classes in Java:

- The wrapper class in Java provides the mechanism to convert primitive into object and object into primitive.
- Java is an object-oriented programming language,we need to deal with objects many times like in Collections, Serialization, Synchronization, etc.

Primitive Type	Wrapper Class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Example:

```
public class WrapperExample1
{
public static void main(String args[])
{
int a=20;
Integer i=Integer.valueOf(a);
Integer j=a;
System.out.println(a+" "+i+" "+j);
}
}
```

Difference between List, Set, and Map in Java

List	Set	Map
The list interface allows duplicate elements	Set does not allow duplicate elements.	The map does not allow duplicate elements
The list maintains insertion order.	Set do not maintain any insertion order.	The map also does not maintain any insertion order.
We can add any number of null values.	But in set almost only one null value.	The map allows a single null key at most and any number of null values.
List implementation classes are Array List, LinkedList.	Set implementation classes are HashSet, LinkedHashSet, and TreeSet.	Map implementation classes are HashMap, HashTable, TreeMap, ConcurrentHashMap, and LinkedHashMap.
The list provides get() method to get the element at a specified index.	Set does not provide get method to get the elements at a specified index	The map does not provide get method to get the elements at a specified index
If you need to access the elements frequently by using the index then we can use the list	If you want to create a collection of unique elements then we can use set	If you want to store the data in the form of key/value pair then we can use the map.
To traverse the list elements by using Listlterator.	Iterator can be used traverse the set elements	Through keyset, value, and entry set.

DATA TYPES IN JAVA:

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

1. **Primitive data types:** The primitive data types include boolean, char, byte, short, int, long, float and double.

2. **Non-primitive data types:** The non-primitive data types include Classes, Interfaces, and Arrays.

Primitive Data Types:

- A primitive data type specifies the size and type of variable values, and it has no additional methods.
- In Java language, primitive data types are the building blocks of data manipulation.

Non-Primitive Data Types:

- Non-primitive data types are called reference types because they refer to objects.
- Examples of non-primitive types are Strings, Arrays, Classes, Interface, etc.