

Name: Sarulatha D

Designation: Trainee

Domain: Testing

## **Java**

### **1. Dictionary in java**

- Abstract class which represents a collection of key-value pairs
- Data structure that stores and retrieves data based on unique keys
- Methods:
  - put(object key, object value) -- adds new key-value pair to dictionary
  - get(object key) -- retrieves value associated with the specified key
  - remove(object key) -- remove the key-value pair with the specified key
  - isEmpty() -- checks if dictionary is empty
  - size() -- returns the number of key-value pairs in dictionary

### **2. Difference between throw and throws keyword**

throw

- Used to explicitly throw an exception from a method or block of code
- Used inside a method
- Syntax: throw new Type("Messages");

throws

- Used in a method signature to declare that method might throw one or more exceptions
- Used when a function might throw an exception during its execution, but it will not resolve the exception rather it ignores the exception
- Should not use throws in programs
- Syntax: public void array (int [] arr) throws ArrayIndexOutOfBoundsException

### **3. What are interfaces and classes**

Classes

- User defined data type
- Blueprint / template that defines the characteristics and behavior of an object
- Class can have constructors used to initialize objects
- Logical entity and consumes no space in memory

Interfaces

- It is an abstract contract that specifies a set of methods that must be implemented by any class that implements the interface
- It is a reference type like class but has only abstract methods
- Able to create variables but default it must be public static final

#### **4. Why interface is base**

- Interface provides a set of methods that can be implemented by multiple classes
- Promoting code reusability, polymorphism
- Implementation can be easily changed without having to change all the references to the functionality

#### **5. Write 15 interface names in java**

- Serializable, Comparable, Runnable, Callable, Iterable, Iterator, Map, Set, Collection, Printable, Clonable, AutoClosable, FunctionalInterface, EventListner, ListIterator.

#### **6. Why all the interfaces end with “able” and what’s the reason**

- The reason for this convention is to make the code more readable. when we see the interface with “able” suffix, we can understand that it defines a capability or a behavior that can be performed by a class that implements interface

#### **7. Graph**

- Nonlinear data structure consisting of nodes or vertices connected by edges.
- Each node contains a value, and each edge may have a label / weight
- Types of graphs
  - Directed graph
  - Undirected graph
  - Weighted graph
  - Un weighted graph
  - Cyclic graph
  - Acyclic graph

#### **8. Difference between HashMap, HashSet, TreeSet**

These are the data structures which store data in different ways.

- HashMap
  - Stores data in key value pairs
  - Keys must be unique, but values can be duplicated
  - Allows one null key
  - No guaranteed order
- HashSet

- Stores only unique elements
- No duplicate elements allowed
- Allows one null element
- No guaranteed order
- TreeSet
  - Stores only unique elements
  - No duplicate elements allowed
  - Does not allow null element
  - Maintains element in sorted order

## **9. Why map is not coming under collections**

- It does not extend collection interface because
- Collection is for a single element storage whereas
- Map is a key-value pair data structure