**PRESENTATION**

**What is class?**

* Class is a blue print (or) template of a object
* Object is a member (also called instance of class)
* There are two types of members
  + - Data member
    - Data member are two types
      * + Static variable refers to the class
        + Non-static variable refers to the object
    - Function member
      * In function member there are
        + Static method
        + Non-static method
* Example

Car is a class in that color, price is a data members and behavior like accelerate, break is a function member

**What is object?**

* Object is an instance of class. it is a real-world entity that exits memory and has a state and behavior
* Creating the object by using class name, reference variable, new keyword and call constructor
* New keyword is used to allocate the memory of an object and the constructor of the class is called to initialize the object
* The constructor you call perform initialization tasks such as initial values for instance variable
* Example

Account is a class has some state like (name, phone no, balance) and behavior like (withdraw and deposit)

We can create an object while create a new customer and storing information of the customer by the help of constructor and we can perform withdraw and deposit function by the help of object name.

**What is reference variable?**

* Reference variable is a variable that holds the memory address of an object rather than the actual memory itself
* Reference variable is used to manipulation of its data and method
* Reference variable that’s declared as class name followed by variable
* Multiple reference variables can point to the same object allowing to shared access and manipulation of object data
* Example

Account is a class has some state like (name, phone no, balance) and behavior like (withdraw and deposit)

We can create an object that object state change the value of phone no by using reference variable

**Static member and non-static member?**

* Static member

Static member is a class member that belongs to the class itself. Static members include static variable and method

* Non-static member

Non-static member belongs to individual object of the class. non-static members include instance variable and method

* Static member (static variable or method) access same class by variable name and method name, access in different class by using class name. variable name and class name. Method name
* Non-static member (non-static variable or method) access same class by using reference variable followed by variable are method, access different class by using instance of object that reference followed by variable or method
* Static members memory allocated once when the class loaded are stored in class area
* Non-static members memory allocated when the object is created are stored in heap area
* There is only one copy of static member its share among all instance of class
* Its own copy of non-static members
* Example for static member

Bank Account is a class that bank name share among all the customer to the same name

* Example
* for non-static members
  + Bank account is a class while customers have own name and phone no

**Member & local variable?**

* Member variable
  + Member variable is a variable declared within a class but outside the method (or) constructor or block is called member variable
  + Scope of member variable depends on the access specifier
  + There are five ways to initialization member variable
    - Direct initialization
    - Using constructor
    - Using blocks
    - Using object reference
    - Using method
    - You not initialize any value it gets default value depends on datatype
* Local variable
  + Local variable is a variable declared inside the Boby of the method is called local variable
  + Their scope is limited to the specific block of code
  + There are few ways to initialization local variable
    - Direct initialization
    - Method call
    - If you not initialize a local variable then complier throw **an error**

**Access specifiers?**

* Access specifiers is a keyword that define the accessibility or visibility of classes, method or variable
* There are four type of access specifier
  + Public
  + Private
  + Protected
  + Default
* Public – when a class, method or variable declare as public we can access on any other class or package.
* Private- private access specifier is a restrict class, method or variable access from its own class.
* Protected-protected access allow a class, method or variable accessed within own package
* Default-if no access specifier is specified the member has default member with default access can only accessed within own package.

**PRE-JAVA**

**What is a programming language?**

* A programming language is a set of rules or instruction that you use to communicate with machine
* Computer only understand machine language which is made up of binary code

**What is java?**

* Java is a high-level, object-oriented programming language developed by sun micro system (now owned by oracle cooperation)
* Java commonly used for developing a web application or standalone application
* Features of java
  + Java is platform independent programming language that can run on any device that has a java virtual machine (JVM) installed, making platform independent
  + Java is based on object-oriented programming which involves the use of classes and objects
  + Java is a robust language, meaning it features strong memory management including compile time checking, garbage collector and exception handling
  + Highly secure, that means java execution is two-step process compilation and interpretation

**Work flow of java?**

* Write the code using a code editor.
* Once the code is written, it needs to be compiled into bytecode using the Java compiler.
* The bytecode is generated and then executed by the JVM. The JVM is responsible for interpreting the bytecode and translating it into machine code.
* At runtime, the JVM loads the bytecode (.class) file and executes the program according to the specified bytecode.

|  |  |  |
| --- | --- | --- |
| JVM | JRE | JDK |
| JVM stands for java virtual machine | JRE stands for java runtime environment | JDK stands for java development kit |
| JVM is a translator with helps us intermediate code to low-level code | JRE provide necessary runtime environment for java programs | Javac converts high-level code to intermediate code |
| JVM has JIT (just in time complier) | JRE has library contains rt.jar file (compressed java files) | JDK contains javac-compiler |
| JVM handles memory management | JRE contains JVM | JDK internally contains JRE |

**What is variable and data type?**

* A variable is a named storage location used to store different type of data during the execution of a program.
* Declare a variable its data-type followed by its name
* A Data type is a pre-defined keyword in java, used to specific type of data we are storing
* There are two types
  + - primitive data type (byte:8-bit, short:16-bit, int:32-bit, float:32-bit, long:64-bit, double:64-bit, char:16-bit, Boolean: represent true or false)
    - Non primitive data type
* Two ways to initialize a variable
  + - Direct initialization
    - Later initialization
* Utilizing the variable involves using their values in various parts of program to perform operations
* Variable there are three types
  + - Member variable
    - Local variable
    - Reference variable

**What is operator?**

* In java, operators are special symbols that are used to perform operations on operands
* Few types of operators in java
  + - Arithmetic operator (used to perform arithmetic operation)
    - Comparison operator (used to compare to value and produce Boolean value)
    - Logical operator (used to perform logic operations on Boolean value)
    - Bitwise operator (used to perform bit-level operation)

**What is flow control statement?**

* Flow control statement is used to control the flow of execution in program
* There are two types
  + - Conditional statement
    - Looping statement
* A conditional statement to execute different block of code based whether specified condition evaluates true or false
  + - There are four types
      * If statement (execute a block of code if a specified condition based on true)
      * If-else statement (execute a block of code if a specified condition true and another block if the condition false)
      * Else-if statement (statement to allow specify multiple conditions)
      * Switch-statement (statement only in case of equal condition)
* Looping-statement is also known as loop it is used to repeat the block of code multiple times
* There are four types
  + - For-loop (repeat the block of code specified number of times)
    - While-loop (repeat the block of code based on Boolean condition)
    - Do-while loop (it always executes the block of code at least once before checking the condition)
    - For-each loop (executing the specified block of code for each element)

**What is function?**

* Function is a block of instruction
* Which can be used to perform some specific task
* Declare a function with an access specifier access modifier return type/void function name with arguments
* Return type refers to data type of value that a method returns after it completes its execution.
* Return-statement return the value form the called method to the calling method
* If a method has non-void return type, then a return type is mandatory

**CORE-JAVA**

**What is encapsulation?**

* Declaring the data member and function member where function members are changing the value of data member is known as encapsulation
* In java encapsulation is achieved by using java bean class
* Encapsulation is used to provide data member security
* Declare a data member as private and through access from getter and setter method
* Getter method return the current value of data type
* Setter method update the value of data member
* For example, in a class like atm the pin number is declare as private allowing access only through setter method thereby ensuring security for that data member

**What is inheritance?**

* One class acquiring the properties of another class is called inheritance
* In java inheritance is achieved by using extends keyword
* The class where the properties are inherited is called super class
* The class which is inheriting the properties is called as subclass
* Using super class object, we can access only super class properties
* Using subclass object, we can access both sub class and super class properties
* Final class or final data member and function member cannot be inherited
* Constructor of the class cannot be inherited
* Private data member and function member of class cannot be inherited
* There are five types of inheritance
  + - Single inheritance
    - Multi-level inheritance
    - Multiple inheritance
    - Hierarchal inheritance
    - Hybrid inheritance

**What is single inheritance?**

* One sub class inheriting the properties of one super class is called single inheritance
* For example, father and son relation

**What is multi-level inheritance?**

* Sub class inheriting the properties of super class and that super class inheriting the properties of another super class is called as multi-level inheritance
* For example, WhatsApp is a class in first version include only text message is in the second version including text message along with the voice message in the third version, expanded it feature to include text message, voice message and video calling

**What is multiple inheritance?**

* Sub class inheriting the properties of two or more super class is called as multiple inheritance
* Java classes doesn’t support multiple inheritance
* It creates ambiguity to the compiler to choose the path from where the properties of object class should be copied to sub class
* To call multiple constructor we should write multiple super statement which is not supported in java
* If both super classes have method with same name and same argument then called using subclass object complier will confuse which method should be called for the Execution

**What is Hierarchal inheritance?**

* Super class inherited extends two or more sub class is called
* For example, the bank account class extends to include both saving account class and current account sub classes, both of which are dependent on bank account super class

**What is Hybrid inheritance?**

* It is combination of different types of inheritance sub classes is called hybrid inheritance
* For example, the Flipkart order is super class extends to further super classes include fashion, electronics and mobiles and these super classes then extends to additional sub classes such as fashion brands extending the fashion class this type of classes structure is known as Hybrid inheritance

**What is abstract class?**

* If a class declared with abstract keyword the class is also known as abstract class
* If a method has only declaration without definition this method is called as abstract method
* Abstract method is developed by using abstract keyword
* If you want to use at least one abstract method then the class is declared as abstract
* We can’t create object of abstract class but we can create the reference variable of abstract class
* We have default constructor in abstract class and also, we can create user-define constructor in abstract class
* We can execute the constructor of an abstract class only by using super statement of subclass

**What is interface?**

* Interface is a java type
* Interface’s default data members are final and static
* Interface default method is abstract
* For every method and variable access specifier by default public
* Impossible to create an object
* There is no constructor & we can’t create user defined constructor
* No relation between interface to object class
* We can create reference variable of interface type
* The classes can inherit the properties of an interface only by using implements keyword
* Where the properties of inherited is called super interface
* Which is implements the properties is called implementation class
* Interface is used to make abstraction, its mean hiding the implementation providing the function to the user by the help of interface
* For example, google map is an interface that includes a function called find location, which provides to Zomato through the interface

**What is type casting?**

* Converting one data type to another data type is called type casting
* There are two types
  + - Primitive casting
    - Derived casting
* Converting one primitive data type of value to another data type of value is called as primitive casting
* There are two types
  + - Widening (converting lower data type value into higher data type value) done by compiler
    - Narrowing (converting higher data type value into lower data type value) using casting statement is explicitly done by programmer
* Converting one object reference type to another object reference type is called as derived casting
* Inheritance is mandatory for derived casting
* There are two types
  + - Up-casting (any superclass reference variable pointing to sub class object is called upcasting)
    - Down-casting (converting up-casted reference back to sub class object reference is called Down-casting)

**What is polymorphism?**

* One entity showing different behavior at different places is called as polymorphism
* There are two types
  + - Runtime-polymorphism
    - Compile-time polymorphism
* Compile time polymorphism
  + - Biding the method declaration to method definition by the compiler of the compile time based on argument is called compile time polymorphism
    - Since the binding is done before the execution is called early binding
    - Once the binding is done it cannot be changed at the runtime hence is called as static binding
    - For example, method overloading
* Runtime polymorphism
  + - Biding the method declaration to method definition by the JVM at the run time based on this object is called as runtime polymorphism
    - Since the binding is done during the execution is called late binding
    - One the binding is done it can be changed at the runtime hence is called as dynamic binding
    - For example, method overriding

**What is abstraction?**

* + Hiding the implements providing the function to the user by the help of interface or abstract class
  + Abstraction is achieved by creating three layers
    1. Object implementation
    2. Object creation
    3. Object utilization
  + Generalize method all the sub class methods and declare them as abstract method in a common super class or super interface
  + Creating object creation layer by creating a class that will create the object of implementation class and upcast into super class or super interface
  + Within the object utilization layer utilize upcasted reference returned by object creation layer to call the generalized method
  + By using abstraction, we can achieve loose coupling where changes done in object implementation layer will not have any impact on object utilization layer

**What is has-a-relation?**

* A relation is a type of relation where two classes are related based on the dependency of existence of one class object over another class object
* Has-a-relation is implemented by creating member reference variable pointing to object of another class
* There are two types of has-a-relation
  + - Aggregation
    - Composition
* Aggregation
  + - Two aggregate objects have own-life-cycle but one object has owner of has-a-relation and child object cannot belong to another parent object
    - For example, A library has a student If the library is destroyed student will exist without library
* Composition
  + - Two composite objects cannot have their own life cycle if one composite object is destroyed, all its parts are also be destroyed
    - For example, a house can have multiple rooms, if house is destroyed all the rooms will be automatically destroyed

**What is singleton class?**

* It is used to create instance of class from same class
* Two steps to create singleton class such as
  + - declare the constructor as private
    - create a static method which returns instance of class object

**Difference between class and abstract class?**

|  |  |
| --- | --- |
| Declare class using class keyword is called class | Declare a class using abstract keyword is called as abstract class |
| We can create the object | We cannot create object |
| Connot written abstract method | We can write abstract method and concert method |
| Execute the constructor by creating object | Execute the constructor by creating sub class object super statement of sub class constructor |

**Difference between interface and abstract class?**

|  |  |
| --- | --- |
| Create the interface by using interface keyword | Create the abstract class by using abstract keyword |
| Variable access specifier can be anything | Variable are default final and static |
| We can write concert method and abstract method | Default abstract method |
| We can write user define constructor | Do have any constructor |
| In-directly inheriting the properties of object class | No relation between object class |

**What is method overriding?**

* + Sub class inheriting the properties of super class and change the method definition according to subclass specification without changing the method declaration is called method overriding
  + Inheritance is mandatory for method overriding
  + Final, private and static method cannot be overridden
  + For example, father have one bike and her son wants to the bike and modify the bike.

**What is method shadowing?**

* In sub class and super class contains static method with same name and same argument is called method shadowing

**What is constructor chaining?**

* + In the process sub class constructor calling super class constructor and super class constructor calling object class constructor is called constructor chaining
  + Constructor chaining is achieved by using super statement
  + It is also known as global constructor chaining
  + It is used to execute the abstract class constructor

**What is constructor overloading?**

* + In same class two or more constructor same name and different arguments is called as constructor overloading
  + It is used to multiple waves to initialize the properties
  + For example, in Instagram multiple waves to create the account using username and password as well as Gamil-id and password

**JAVA-library**

**What is object class?**

* Object class is an inbuild class in java
* It presents in java. Lang package
* Object class is a super most class in java directly are indirectly all class are inheriting the properties
* There is few in-build method present in object class, such as
  + - Get class ()
    - Has code ()
    - To-string ()
    - Equals ()
    - Wait ()
    - Wait (long)
    - Wait (long, int)
    - Notify ()
    - Notify all ()
    - Object clone ()
    - Finalize ()

**What is string class?**

* String is an inbuilt class in java
* Which is used to store group of character the string object
* String class is immediate sub-class of object class
* String class is final cannot be modify
* Has-code, to-string and equals methods of object class overridden by the string class
* String class implements serializable, comparable <string> and character sequence
* String objects are created by using new keyword and without new keyword
* String object are created in special memory called string pool inside heap area
* String poll consist two parts
  + - Constants poll (duplicates are not allowed)
    - Non-constant poll (duplicates are not allowed
* For example, string s1=” hello” string s2=” hello” two reference are pointing to same object, string s1=new string (“hello”) and string s2=new string (“hello”) two reference are pointing to different objects

**Why string is immutable?**

* If we try to re-initialize a string object then a new object will be created with the new value of reference variable start pointing to new object leaving the old object the de-reference variable object
* For example, string s1= “hello” then s1=s1. Concat (“java”)

**Difference between string class and object class?**

|  |  |
| --- | --- |
| Object class hash-code method return unique int value generated by JVM based on address of an object | String class hash-code methods return unique value based on given object character |
| To-string method represent fully qualified name and @ and hash-code value | To-string method return current value of an object |
| Equal method compares hash code vale generated JVM based on address | Equal method compares hash code value generated based on unique value |

**Difference between string and string buffer?**

|  |  |
| --- | --- |
| String object are immutable | String buffer objects are mutable |
| String object can be created by using new keyword or without new keyword | String buffer object created by using only new keyword |
| Hash-code, to-string, equals methods of object class overridden by string class | Only to-string method of object class overridden by string buffer class |
| String objects stored in special memory called string pool | There is no special memory |
| Concate methos support in string class | Instead of concatenation we need to use append method in string buffer class |
| String class is not a thread safe class | Is a thread save class |

**Difference between string buffer and string builder**?

* there is only one difference between string buffer and string builder
  + - string buffer is a thread safe class
    - string builder is a not thread safe class

**difference between string class and string builder class?**

|  |  |
| --- | --- |
| String object are immutable | String builder objects are mutable |
| String object can be created by using new keyword or without new keyword | String builder object created by using only new keyword |
| Hash-code, to-string, equals methods of object class overridden by string class | Only to-string method of object class overridden by string builder class |
| String objects stored in special memory called string pool | There is no special memory |
| Concate methos support in string class | Instead of concatenation we need to use append method in string builder class |

**what is wrapper class?**

* + Converting primitive into java object is known as wrapper class
  + There are two types
    1. Autoboxing
    2. Unboxing
  + Autoboxing
    1. There is a mechanism converting primitive into object
  + Unboxing
    1. There is a mechanism converting java object into primitives
  + Every wrapper class can be upcasted into object class
  + For every primitive data type, we have one corresponding wrapper classes
  + For example, char data type has one wrapper class is called as character

**What is exception handling?**

* + Exception is an unexpected event which occurs as run-time due to unexpected operation perform by the single line of code
  + There are two types of exception
    - 1. Checked exception
      2. Un-checked exception
  + Checked exception
    1. Leaving run time exception and its sub classes all other exception is called checked exception
  + Un-checked exception
    1. Exception which are not checked by the compiler at the compile time are called an unchecked exception
  + Where ever there is an exception JVM will come and create an object of corresponding exception class
  + JVM will pass the exception object to the method which is create the exception by using throw keyword
  + If the method is not able to handle the exception object, then JVM will terminate the methods execution
  + If no method is handling the exception object the JVM will call default exception handler which will handle the exception object
  + Default exception handler will handle the exception and print
    - 1. Name of the exception
      2. Reason of the exception
      3. Complete stack trace

**How to handle exception?**

* + Exception can be handled by using try and catch block
  + Try block is used to write risky line of code
  + Risky line of code is nothing but is code is reason for exception
  + Try block throw the exception to a catch block
  + Catch block is used to catch the exception object which is throw by the try block and execute alternate code instead of exception

**What is throws keyword?**

* + Throws keyword is used to propagation checked exception explicitly by the programmer

**What is throw keyword**?

* + Throw is used to throw the exception explicitly by the programmer according to the application requirement

**What is customized exception?**

* + Creating user defined exception based on application requirement is called as customized exception
  + Few steps to create customized exception such as
    - 1. Create a class which extend exception class
      2. Create an object of your customized exception by using throw keyword and handle by with try and catch block
      3. If you want to make customized exception class as checked exception and need to extends exception class
      4. If you want to make customized exception class as un-checked exception then need to extend runtime exception of sub class

**What is thread?**

* + A thread is an independent part of some program which will get in own stack and cup time for the execution
  + Thread creation and execution is a costly resource
  + Two ways to create thread such as
    - 1. By extends thread class
      2. By implements runnable interface
  + Thread is used to perform multitasking in a program or application
  + Whenever JVM start the execution, it creates 3 threads by default
    1. Main thread
    2. Garbage collector
    3. Thread scheduler
  + By default, all the programs in java will be executed in main thread

**How to create thread in java?**

* + There are two ways to created thread in java
    - 1. By extends thread class
      2. By implements runnable interface
  + Extends thread class
    - 1. Following few steps to create and execute
         1. Create a class extends thread class
         2. Override run method to define the task should be executed by the thread
         3. Create the object of subclass
         4. Use the sub class object to call start method of thread class
  + Implements runnable interface
    - 1. Following few steps to create and execute
         1. Create a class that implements runnable interface
         2. Override run method to define the task should be executed by the thread
         3. Create the object of implementation class
         4. Create the object of thread class and pass the implementation class object reference to thread constructor
         5. Use the thread class object to call start method of thread class

**What are the methods in thread class?**

* + There are some methods in thread class
    - 1. Start ()- it is used to start the execution
      2. Get id ()-it used to return the id of the thread
      3. Get name ()-it is used to return the name of a thread
      4. Get priority ()-it is used to return priority of thread
      5. Set name ()-it is used to set the name of thread
      6. Set priority ()-it is used to set the priority of the thread

**What is multithreading?**

* + Multithreading is the process of execution multiple threads at same time
  + That allows concurrent execution of two or more parts of a program for maximum initialization cup

**What is race condition?**

* + Multiple threads trying to access same object at same time is called as race condition
  + Race condition always bad to inconsistent data
  + Race condition can by overcome by using thread synchronization
  + Thread synchronization is used to execute thread in sequential order where one thread will execute only after other thread complete the execution
  + Thread synchronization can be achieved by two ways
    - 1. Using synchronized method (or)
      2. Using synchronized block
  + For example, consider book is a synchronized method or synchronized block, one person read the book and give to another then person will be read

**What is thread dead lock?**

* + It is situation where two thread are waiting to lock the object of each other to complete the execution and wait for infinite period of time
  + Thread dead lock can be overcome by using inter thread communication (ITC)
  + Inter thread communication achieved by using methods of object class
    - 1. Wait ()-it will pause the execution of the thread and release all the object lock held by the using given object
      2. Notify ()-it will send the notification in the thread which is in wait state to resume execution
      3. Notify all ()- it will send the notification to all thread which are in wait state to resume its execution

**Explain thread life cycle?**

* + New state-when new thread is created it is a new state
  + Runnable
    - 1. A thread is read to run that’s move to runnable state
      2. In this state thread might be ready to run or might be running at instance of time
  + Waiting or blocking
    - 1. When a thread is inactive then its is one of the following states
         1. Wait state
         2. Block state
  + Thread waiting
    1. A thread lies in waiting state when it called method with a time out
    2. A thread lies in this state util the time out is completed or util a notification is received
  + Terminated (dead state)
    - 1. Thread has completed its execution

**What is collection?**

* + Collection is an interface
  + Its present in java. Util package
  + Collection is a group of reference which is represent as one unit
  + Collection is three types
    1. List
    2. Set
    3. Queue

**What are the methods in collection?**

* + There are some methods in their collection such as
    - 1. Add ()-it is used to insert the element
      2. Add (index, object)-it is used to insert based on index
      3. Add all (collection c)-used to insert complete collection
      4. Add all (index, collection)-used to insert complete collection based on index
      5. Get (index)-used to retrieve element based on index
      6. Size ()-used to calculate the size of element
      7. Remove (index)-used to remove element based on index
      8. Remove all (collection)-used to remove all element
      9. Clear ()-used to remove complete element
      10. Contains (object)-used to check element present or not
      11. Contains all (collection)-used to check complete collection is present or not

**What is list?**

* + List interface is a sub interface of collection interface which has some properties
    - 1. List can store duplicate values
      2. List has index
      3. List support multiple null values
      4. List preserve insertion order

**What is array list?**

* + It is an Implementation class of list interface
  + Array list can store duplicates value
  + Array list has index
  + Array list support multiple null value
  + Array list preserve insertion order
  + Underlying data structure of Array list is resizable array (co.\*3/2+1)
  + Add the element to Array list by using add ()
  + Retrieve the element from Array list by using get ()
  + Array list have three constructors
    - 1. Array list ()-constructor an empty list with an initial capacity 10
      2. Array list (collection c)- constructor a list contains the elements to specified collection
      3. Array list (int initial capacity)-constructor an empty list with the specified initial capacity

**What is vector list?**

* + It is an Implementation class of list interface
  + Vector list is a thread safe class because its implements
  + Vector list can store duplicates value
  + Vector list has index
  + Vector list support multiple null value
  + Vector list preserve insertion order
  + Underlying data structure of Vector list is resizable array (co.\*2)
  + Add the element to Vector list by using add ()
  + Retrieve the element from Vector list by using get ()
  + Array list have four constructors
    - 1. Vector list ()-constructor an empty list with an initial capacity 10
      2. Vector list (collection c)- constructor a list contains the elements to specified collection
      3. Vector list (int initial capacity)-constructor an empty list with the specified initial capacity
      4. Vector list (initial capacity, capacity increment)-constructor an empty list with the specified initial capacity and specified capacity increment

**Difference between array list and vector list?**

|  |  |
| --- | --- |
| Array list is not implements synchronized | Vector list is implements synchronized |
| Array list is not thread safe class | Vector list is thread safe class |
| Multiple threads are accessing one thread modify the list automatically it will impact on another thread | Multiple threads are accessing one thread modify the list structure is will not impact another thread |
| Array list cannot be specified capacity increment | Vector list can be specified capacity increment by using constructor |
| Resizable in current capacity\*3/2+1 | Resizable in current capacity\*2 |

**What is set?**

* + set interface is a sub interface of collection interface which has some properties
    - 1. set do not store duplicate values
      2. set do not have index
      3. set support to store only one null values
      4. set do not preserve insertion order

**what is hash set?**

* + It is an Implementation class of set interface
  + hash set do not store duplicates value
  + hash set do not have index
  + hash set support to store only one null value
  + hash set do not preserve insertion order
  + Underlying data structure of hash set is hash table
  + Add the element to hash set by using add ()
  + Retrieve the element from hash set by using get ()

**What is linked hash set?**

* + It is an Implementation class of set interface
  + Linked hash set do not store duplicates value
  + Linked hash set do not have index
  + Linked hash set support to store only one null value
  + Linked hash set always preserve insertion order
  + Underlying data structure of linked hash set is LinkedList
  + Add the element to linked hash set by using add ()
  + Retrieve the element from linked hash set by using get ()
  + Linked hash set have four constructors
    - 1. Linked hash set ()-constructor an empty list with an initial capacity 10 and load factor (0.75)
      2. Linked hash set (collection c)- constructor a list contains the elements to specified collection
      3. Linked hash set (int initial capacity)-constructor an empty list with the specified initial capacity and default capacity (0.75)
      4. Linked hash set (initial capacity, float factor)-constructor an empty list with the specified initial capacity and specified load factor

**Difference between has-set and linked has-set?**

|  |  |
| --- | --- |
| Has-set does not preserve insertion order | Linked-hah-set always preserve insertion order |
| Under-lying data structure of has-set is hash table | Under-lying data structure of linked has-set is linked list |

**What is tree set?**

* + Tree set is an implementation class of set interface
  + It is a non-linear data structure where data is stored according to some relation
  + Trees consists node and list
  + Node consist data and address of other data
  + The link represents the relation between one node to another node
  + Tress set do not store duplicates
  + Tree set don’t have index
  + Tree set do not support one null values (if we try to add one null value JVM will throw null pointer exception)
  + Tree set do not preserve insertion order
  + Tree set do not allow heterogenous data
  + Underlying data structure of tree set is binary search
  + There are four constructors in tree set
    - 1. Default constructor is natural sorting order
      2. Comparator c-where the elements will be inserted according to customized sorting order
      3. Collection c
      4. Sorted set

**What is queue?**

* + Queue is a sub interface of collection interface which have some properties
  + Queue store duplicates
  + Queue has index
  + Queue support to store multiple null values
  + Queue preserve insertion order

**What is priority queue?**

* + Priority queue store duplicates
  + Priority queue doesn’t have index
  + Priority queue does not support one null values (if we try to add one null value JVM will throw null pointer exception)
  + Priority queue does not preserve insertion order
  + Priority queue does not allow heterogenous data
  + Underlying data structure of Priority queue is Priority heap
  + Data from Priority queue retrieve elements by using poll ()
  + Pool method retrieve and remove the head element of queue
  + There are four constructors in tree set
    - 1. Default constructor is natural sorting order
      2. Comparator c-where the elements will be inserted according to customized sorting order
      3. Initial capacity -the specified capacity that order elements according to their natural sorting
      4. Initial capacity and comparator c- the specified capacity that order elements according to their customized sorting

**What is linked list?**

* + Linked list is an implementation class of list and queue interface
  + Linked list store duplicates value
  + Linked list has index
  + Linked list preserve insertion order
  + Underlying data structure of Linked list is doubly Linked list
  + Data from Linked list to retrieve using poll method and other methods

**Difference between tree set and priority queue?**

|  |  |
| --- | --- |
| Tree set is a non-linear data structure | Priority queue is a linear data structure |
| Retrieve by elements by using get method | Retrieve by elements by using poll method |
| Tress set not specified capacity increment | Priority queue specified the capacity increment |
| Underlying data structure of tree set is binary search | Underlying data structure of priority queue is priority heap |

**Difference between linked has set and linked list?**

|  |  |
| --- | --- |
| Linked has set do not store duplicates value | Linked list store duplicates value |
| Linked has set do not have index | It has index |
| It supports to store only one null value | It supports to store multiple null value |
| It does not support poll method | It supports poll method |
| Underlying data structure of linked has set is linked list | Underlying data structure of linked list is doubly linked list |

**What is map?**

* + In java map is an interface that represent a collection of key value pairs where each key is unique and is implemented by hash-map, liked hap-map, tree-map and others

**What is hash-map?**

* + Hash-map is the implementation class of map interface
  + Do not store duplicate key
  + It can be storing duplicate values
  + Hash map does not have index
  + Key and values can de added into has-map by using put ()
  + To retrieve by using get (key)
  + If we try to add new value into old key then old key will be replaced the new value of given key
  + Only one key can be null
  + Values in hash-map can be null
  + Does not preserve insertion order
  + Underlying data structure of hap-map is hash table

**What is linked hash-map?**

* + Linked Hash-map is the implementation class of map interface
  + Do not store duplicate key
  + It can be storing duplicate values
  + Linked Hash map does not have index
  + Key and values can de added into Linked has-map by using put ()
  + To retrieve by using get (key)
  + If we try to add new value into old key then old key will be replaced the new value of given key
  + Only one key can be null
  + Values in Linked hash-map can be null
  + Do preserve insertion order
  + Underlying data structure of Linked hap-map is Linked list and hash table

**Difference between hash-map and linked has-map?**

|  |  |
| --- | --- |
| Hash-map does not preserve insertion order | Linked hap map preserve insertion order |
| Underlying data structure of hap-map is hash table | Underlying data structure of Linked hap-map is Linked list and hash table |

**What is tree map?**

* + Tree-map is the implementation class of map interface
  + Do not store duplicate key
  + It can be storing duplicate values
  + Tree map does not have index
  + Key and values can de added into Tree map by using put ()
  + To retrieve by using get (key)
  + If we try to add new value into old key then old key will be replaced the new value of given key
  + Null keys cannot be allowed
  + Values in Tree -map can be null
  + Does not preserve insertion order
  + Underlying data structure of Tree -map is binary search
  + Tree map should be key homogonous
  + Values of tree map can be heterogonous
  + The values in tree map are store according to key

**Difference between tree map and hash map?**

|  |  |
| --- | --- |
| In tree map key are only homogonous | In hash map keys are heterogonous |
| Null keys cannot be allowed | Only one null key allowed |
| Underlying data structure of Tree -map is binary search | Underlying data structure of hap-map is hash table |

**Difference between tree map and hash map?**

|  |  |
| --- | --- |
| In tree map key are only homogonous | In hash map keys are heterogonous |
| Null keys cannot be allowed | Only one null key allowed |
| Underlying data structure of Tree -map is binary search | Underlying data structure of hap-map is hash table |

**Difference between tree map and linked has map?**

|  |  |
| --- | --- |
| In tree map key are only homogonous | In linked hash map keys are heterogonous |
| Null keys cannot be allowed | Only one null key allowed |
| Does not preserve insertion order | Do preserve insertion order |
| Underlying data structure of Tree -map is binary search | Underlying data structure of linked hap-map is linked list and hash table |

**Difference between hash-map and has set?**

|  |  |
| --- | --- |
| Hash map implementation class of map interface | Hash set implementation class of set interface |
| Hash map stored the data key and value pairs | Hash set stored the data only values |
| One or more null value can be allowed but only one null key allowed | Only one null value can be allowed |
| Add the elements from hash map by using put method | Add the elements from has set by using add method |

**Difference between linked hash map and linked list?**

|  |  |
| --- | --- |
| Linked Hash map implementation class of map interface | Linked list implementation class of set and queue interface |
| Linked Hash map stored the data key and value pairs | Linked list stored the data only values |
| Add the elements from hash map by using put method | Add the elements from has set by using add method |
| Underlying data structure of Linked hash map is linked list and hash table | Underlying data structure of Linked list is doubly linked list |
| It does not have index | It has index |

**Difference between tree set and tree map?**

|  |  |
| --- | --- |
| Tree set implementation class of set interface | Tree map implementation class of map interface |
| Tree stored the data only values | Tree map stored the data key and value pairs |
| Add the elements from has set by using add method | Add the elements from hash map by using put method |
| Only homogonous values are allowed | Both homogonous and heterogonous values are allowed |

**What is iterator?**

* Iterator is a class present in java. Util package
* Iterator is a cursor which is used to retrieve the elements present in given collection
* Iterator is created automatically whenever the collection object is created
* For every collection sperate iterator are created
* We can access iterator of given collection by using iterator ()
* The iterator reference returned by the iterator method should be stored with a reference of iterator type

**Why do we need iterator?**

* Iterator is used to retrieve the all elements present in the given collection

**Why do we need generic?**

* Generic is used to perform two major operations
  + - To make given collection type safe
    - To avoid unwanted down casting

**What is comparable?**

* Comparable is an interface present in java. Lang package
* Comparable interface contains only one method compare to
* Return type of compare to method is int
* Compare to method overridden by all wrapper classes and string class
* It is used to default natural sorting

**Why do we need comparable?**

* Comparable is used to provide natural sorting
* To utilize the comparable interface in collection, use the sort method provide by the collections class such as collections. Sort ()
* When you call the collections. Sort method it internally uses the compare to method

**What is comparator?**

* Comparator is an interface present in java. Util package
* Comparator interface contain only one method compare method
* Return type of compare method is int
* It is used to provide customized sorting in the object
* To use comparator interface, we have to follow four steps
  + - Create the class that implements comparator interface
    - Override the compare method and specific your customized logic to sort object
    - Create the implementation class object
    - Pass the implementation class object reference in the constructor of tree set

**Why do we need comparator?**

* Comparator is used to provide customized sorting
* To use comparator interface, we have to follow four steps
  + - Create the class that implements comparator interface
    - Override the compare method and specific your customized logic to sort object
    - Create the implementation class object
    - Pass the implementation class object reference in the constructor of tree set

**Difference between comparable and comparator?**

|  |  |
| --- | --- |
| Comparable interface present in java. Lang package | Comparator interface present in java. Util package |
| It contains only one method compare to method | It contains only one method compare method |
| The actual class it modified comparable affect original class | It does not affect original class |
| We sort the elements in collections. sort (list) | We sort the elements in collections. Sort (list, compare c) |

**Difference between collection and collections?**

|  |  |
| --- | --- |
| Collection is an interface | Collections is a class |
| It is used to represent group of individual objects act a single unit | It is used to perform on collection by using utility methods |
| Collection interface that contains abstract method since java 8 the interface can also contain static and defaults method | It contains only static methos |

**What is garbage collection?**

* It is used to remove the unused object from heap area to garbage collector
* To make object as unused we should re-initialize that reference to null
* In java process will be done automatically to keep the memory clean
* If programmer want to remove the object from heap area to garbage collector than we need to use System. Gc ()

**What is finalize method?**

* It is a method of object class which execute automatically just before the object is removed from heap area

**Difference between final and finally?**

|  |  |
| --- | --- |
| Final is a keyword in java | Finally, is a block |
| Final is used to restrict the variable, class and methods | Finally, block is executing irrespective Occurrence of execution |

**Difference between finally and finalize?**

|  |  |
| --- | --- |
| Finally, is a block | It is a method of object class |
| Finally, block is executing irrespective Occurrence of execution | Finalize method execute automatically just before the object is removed from heap area |

**What is file handling?**

* File handling is the process of reading from and writing to file on the file system using java code
* The operation can be performed on a file
  + - Create a file
      * Create a file by using create file class object, its present in java. Io package
    - Write to file
      * By using writer class, we can write data to file
    - Read to file
      * By using scanner class, file reader class and buffer reader class we can read data from file
    - Delete a file
      * To delete a file, we can use file object delete method

**How to create a file in java?**

* Create a file by using file class object file class present in java. Io package

**How to write to a file?**

* File writer class used to write data to file
* File writer class write method is used to write data to file
* Close method is used to the file after close we can write anything to a file or we can use flush method

**What is stream in java?**

* In java sequence of data is known as stream
* This concept is used to perform input and output operation on a file
* There are two types of streams
  + Input streams
    - File input stream
    - Object input stream
  + Output streams
    - File output stream
    - Object output stream

**What is serialization?**

* The process of writing state of object to a file is called serialization
* It is the process of converting an object from java supported form to a file supported form or network supported
* By using file output stream and object output stream we can achieved serialization

**What is de- serialization?**

* The process of reading state of object from to a file is called de- serialization
* In the process of converting an object form file supported form or network supported form in java supported form
* By using file input stream and object input stream classes we can achieved de- serialization

**What is transient keyword in java?**

* Transient mean not to serializable
* It is a modifier which is applicable only for variable
* In serialization if we don’t want to save the value of variable for security the we should use transient keyword
* If variable in transient the JVM ignore the original value of transient variable and save the default value of the