**Tell me about yourself ?**

**=>** Good afternoon sir , first of all thanks for giving me this opportunity .

Sir , my name is shubham devidas dhumal , I am from pune maharastra ,

Recetly I completed my post graduation diploma in advance computing .

During my studies i gain knoweldge on programming technology and also I have worked on sevral projects . like student managmnt system , employee management system etc . and also completed travel and tourism management system as acemdeic project .

Sir I am really excited about this position because I am passioniate about problem solving and software devlopment . I belive that my skills align well with this job opertunity .

Thank you for consdring my application .

**What is java ?**

**=>** Java is high level , object-oriented programming language devloped by sun-microsysytems .

Is designed to be platform independent that means java code can run on on any platform that has JVM installed .

Java is used to devlop web applications , enterprize applications , mobile applications .

**What is oops ?**

**=>** oop stands for object oriented programming . is a way of programming .

Oops is a programming methodology which organize complex problem with the help of Inheritance , encapsulation , polymorphism and abstraction .

**What is class ?**

**=>** class is template use to create object . or class is blue print which define objects of certain kinds .class is logical entity contains properties and behaviours .

Properties means variable and behaviours means methods .

For example : car , book , student

Suppose I have class student which has some properties like student name , roll no , address but don’t have any values associated with .

**What is object ?**

**=>** object is real world entity which has state , behavior and identity .

State means object has some data associated with it .

Behavior means it exhibit some actions .

Identity means it should be uniquly identifyble .

It is an instance of a class .In java we create objects by using new keword . and whenever we create objects , every instance members push into heap memory . and we can use them by using reference variable .

Example : my car , red book , ramesh student.

**What is Encapsulation ?**

**=>** encapsulation means binding data together into a single unit .

Lets have rel life example there is word capsule , what capsule is doing , it wrapping data into single unit and provide protection to it by its outer cover .

So ,Similarly with programming language class is best example of encapsulation .

Class wraps data member and member functions into single unit and we provide protection using access modifirs like private and protected .

So , class is best way to bind data into single unit that is encapsulation .

**What is abstarction ?**

**=>** abstraction is one of the object oriented principle in java .

Abstaction is an act of represnting essential features and hiding internal implimentations details .

For example : I have one Account class , in that I have getBalance() method .

Here I will never highlights this method to end user , instead I give GUI screen and provide check balance button when user click this button that getBalance() method will execute .

In this way the user don’t know which code is executing he only see button .

That is abstaction , hide internal implimentation details and only showing essentials .

**Is abstract method can be inherit ?**

**=>** yes we have to inherit abstract method . because whenever we declare a class as abstract we cannot create any instance o It . we have to extend it in another class .

**What is inheritance ?**

**=>** when we want to extend or reuse properties of parent class into child class we use inheritance .

For exaple :

Suppose I have class A and it has data members like name , emailid and class B , has same data members like name , emailid and one extra data memebr like address so I don’t required to write code again , I just need to extend class A . like

Public class B extends A { }

In this way all data members of class A will inherited to class B.

This is possible because of inheritance .

**What is polymorphism ?**

**=>** in polymorphism , poly means many and morph means forms ie. Having multiple forms is nothing but polymorphism .

For example : a person in class is a student and same person in market is a customer

Here person is same but its behavior is different . so we can say that it is polymorphism .

In java we have two types of polymorphism , compile time polymorphism and run time polymorphism

Method overloading is the example of compile time polymorphism .

And method overriding is the example of run time polymorphism .

**What is method overloading ?**

**=>** method overloading is the example of compile time polymorphism.

We can achieved method overloading by providing multiple methods with the same name but with different parameters in a same class . we can achieved it by changing numbers of parameters , or by changing data types of parametrs

**What is methodoverriding ?**

**=>** Method overriding is the example of run time polymorphism .

Method overriding allow a subclass to provide specific implimentations of a method that is already defined in superclass .

**What is compile time polymorphism and run time polymorphism ?**

**=>**

method overloading is the example of compile time polymorphism . in Compile time polymorphism compiler determine which method has to call before the execution of program on the basis of number and type of arguments passed to the method .

Run time polymorphism means the decision of which method has to call is made at runtime on the basis of on which object the method is call .

**By inheriting a class what actually get inherited ?**

**=>** public ,default , protected members variables and methods of the class only gets inherited .

Private memebr variables and methods of the class is not inherited .

**What is multithreading?**

**=>** multi threading is a process of executing multiple threads parallely in a single process .thread is lightweight process that can be run parallay with other threads within a same process .

We can create threads in two ways :

By extending thread class and overriding run method or

By implimenting runnable interface and passing instance of class to thread constructor.

**What is thread ?**

**=>** thread is lightweight process that can be run parallay with other threads within a same process .

**Explain life cycle of thread ?**

**=>**

**Explain join method ?**

**=>** join method wait for other thread to complete its execution . suppose we have two threads t1 and t2 . and if running thread t1 invoke join on t2 then t1 thread will wait into waiting state util t2 completes execution . then t1 completes its execution .

**Explain sleep() ?**

**=>** sleep() method causes running thread to sleep for specified amount of time .

**Explain wait() , notify() , and notifyAll() ?**

**=>** wait() method causes the current thread to wait until another thread invokes notify() or notifyAll() for that object.

Notify() method wakes up single thread that is waiting .

And notifyAll() method wakes up all thread that are waiting .

**What is deadlock in multithreading ?**

**=>** Deadlock is situation when two or more threads are blocked and waiting for each other to release a resource ,because of this situation none of threads are processing .

**What is synchronization in multithreading?**

=> Synchronization is the process of co-ordinating multiple threads ,

When two or more threads try to access common resource concurrently , they may leads to data inconsistency , to avoid such problems we use synchronization .

Synchronization ensurs that only one thread can access common resource at a time.

**What is a race condition in multithreading?**

**=>** race condition occurs when multiple threads trying to access common resource simultaneously ,because of this we get unpredictable data. Means That time threads do not co-oridanate with each other resulting in race condition .

**===========================================================================**

**What is exception ? and how to handle it ?**

**=>** exception is an unexpected event that affects normal flow of execution of program.

exception can occurs at both compile time and run time

We can handle exception by using exception handling techniques like by using

Try-catch block

Or by using throws keyword .

**Explain trow and throws keyword ?**

**=>** throw keyword is used to explicitly throw an exception from method . throw keyword stops the normal flow of execution .

For example suppose I have class calculator and I have method division in it which will take number from user by which we going to divide another number , and in that I write if condition like if that number == 0 , then I will throw exception by using throw keyword and write gracefull message means like

throw new ArithmaticException(“cannot divide by zero”);

Or user enters string instead of number I will throw numberformatexception

Throws keyword is used in method declaration . to declare method that can potentially throw an exception .

For example : if I write method that reads data from a file, and it may happen the file may not exist or may not be readable that time I will write :

public void readfile() throws IOexception { }

Mens this method can throw an exception if method not executed successfully.

[ note :

Error : error class is the subclass of Throwable class . it terminates the program if there is a error .

Exception : exception is subclass of throwable class

]

**What is Checked exceptions**

=> checked exception are checked by java compiler at compile time . for example checked exceptions like class notfound exception , SQLexception , FileNotFound exception etc

It must be handle using try-catch block or throws keyword .

**What is Unchecked exception?**

**=>** unchecked exception are also called as runtime exception , that are not checked at compile time.

Unchecked Exeption like ArithmaticException , ArrayIndexOutOfBoundsException , etc

**What is an Error ?**

**=>** Errors represent irrecovarable conditions , it usally indiacates problem with the environment or sysytem .

examples of errors in java include outOfMemory Error (unable to allocate enough memory) , StackOverFlowError(programs call stacks exceeds max size) and VirtualMachineError.(problem with JVM)

**Difference between Exception and Error ?**

**=>** basically exception and error both are subclass of the throwable class .

Error represents irrecovarable conditions ,it usally indicates problem with the system environment .

Whereas

Exception is unexpected event that affesct normal flow of execution of program .

Exceptions are recovarable .

Errors occurs at runtime only

While exception can occurs at both compile time and run time .

Example of errors are outOfMemory , StackOverFlow

Wheras example of exceptions are NullPointer Exceptions , ArrayIndexOutOfBonds Exception.

**What is collection framework ?**

**=>** collection framework consists of classes and interfaces that are used to store and maintain groups of objects .

It includes classes like ArrayList , LinkedList , HashSet , TreeSet , and HashMap and TreeMap

And interfaces like List , Map , and Set

**What is difference between List , Set and Map interface ?**

**=>** List interface is in collection framefork , which is used to store groups of objects ,

List interface maintains insertion order of elements and allow duplicates elements also .

Set interface is in collection framework , which is used to store group of objects ,

Set interface does not maintains insertion order of elements and store only unique elements , means duplicates are not allowed in set .

Map interface is in collection framework , which is used to store group of objects .

MAP store elements in a key-value pair , where keys are unique and assosiated with particular value .

We can access values based on their keys.

**What is difference between array and arrayList ?**

**=>** Array is an data structure , which allows us to store multiple values of same data type in a single variable.

Elements are arranged in contigious memory location and each elements are identified by an index number which starts from 0.

ArrayList is a class in collection framework which impliments List interface .means it maintains insertion order and also allow duplicate elements.

ArrayList is a dynamic array which can grow or shrink in size as needed . Arraylist can store elements of different data type .

|  |  |
| --- | --- |
| Array | ArrayList |
| Array is fixed size container . size of the array cannot change once it initialsied. | Whereas Arraylist is dynamic . which can grow or shrink in size as needed. |
| Array can store elements primitive data as well as objects. | Whereas arraylist store objects but it cannot store data of primitive types. |
| Array can be multip dimensional  Integer addarrayobject[][] = new Integer[3][2]; | While arraylist is single dimensional . |

**What is difference between hashset and hashmap?**

**=>** HashMap : hashMap is a class in collection framework which impliments Map interface.

Means elements are stored in a key-value pairs in hashMap . Where keys are unique and assosiated with particular value .

HashMap does not maintain insertion order [, because hashmap uses hash table to store elements and insertion depends on hash function used by hashMap. ]

We can access values based on their keys . it can accepts one null key and accept multiple null values .

HashSet : hashset is a class in collection framework which impliments Set interface . means it doesnot maintains insertion order and only unique elements are allowed in a set , and duplicates are not allowed .

Elements are added into HashSet based on there hash code .if two elements have same hashcode then they are considered equal and only one element is added.

|  |  |
| --- | --- |
| Hashmap | hashset |
| hashMap is a class in collection framework which impliments Map interface | hashset is a class in collection framework which impliments Set interface |
| In hashmap elements are stored in key value pairs | In hashset we store objects. |
| Hashmap does not maintains insertion order and it accepts only uniqe keys . | Hashset does not maintain insertion order and does not allow duplicates. |
|  |  |

**What is purpose of TreeSet and TreeMap ?**

**=>** TreeMap is a class in collection framework which impliments SortedMap interface which extends Map interface . means elements are stored in a key-value pairs in treemap .

Treemap maintains elements in a sorted oreder according to there keys .Where keys are unique and assosiated with particular value .

TreeSet is class in collection framewok which impliments SortedSet interface which extends from Set interface .means only unique elements are allowed in a treeset and and TreeSet maintains elements in sorted order .

**Which class is used to maintain insertion order and does not allow duplicates ?**

**=>** LinkedHashSet calss is used to maintain insertion order and does not allow duplicates .

Is a subclass of HashSet that maintains doubly linked list of the elements in the order in which they inserted .

**Difference between linkedHashMap and LinkedHashSet ?**

**=>** LinkedHashSet is used when we want to maintain insertion of order and store only unique elements .

Whereas LinkedHashMap is used when we want to store elements in key -value pair and it also maintains insertion order .

**What is the purpose of the Collection interface?**

The Collection interface is the root interface of the Collection framework, and provides a way to manipulate groups of objects. It defines common methods like add(), remove(), size(), and contains(), that are shared by all Collection classes.

**What is the purpose of iterator interface?**

**=>** iterator interface provides standard way to access and process each element in the collection . iterator interface has methods like :

hasNext() which return true if iterator has more elements to iterate otherwise it return false.

next() method returns next element in collection .

[remove() method removes last element returned by the iterator ]

**What is the difference between Comparable and Comparator interfaces?**

Comparable and Comparator are both interfaces used for sorting objects, but they have different purposes. Comparable is used to provide a natural ordering for objects, based on their inherent properties. Comparator is used to provide a custom ordering for objects, based on specific criteria.

**What is the purpose of the Queue interface?**

The Queue interface is used to store a collection of elements in a particular order, where the elements can be inserted at one end and removed from the other end. It provides methods like add(), remove(), peek(), and offer(), that allow you to manipulate the elements of the queue.

**What is the purpose of the Deque interface?**

The Deque (short for "double-ended queue") interface is used to store a collection of elements in a particular order, where elements can be inserted and removed from both ends of the deque. It provides methods like addFirst(), addLast(), removeFirst(), and removeLast(), that allow you to manipulate the elements of the deque.

**What is an array?**

**=>** Array is an data structure , which allows us to store multiple values of same data type in a single variable.

Elements are arranged in contigious memory location and each elements are identified by an index number which starts from 0.

We can create array by two ways :

Declare an array and initialize it later :

Int[] arr = new int[5]

arr[0] = 1;

arr[1] = 2;

arr[2] =3;

Declare and initialise array in a single statement

Int[] arr = { 1 , 2 , 3 , 4 , 5}

**What is interface ?**

**=>** Interface is blueprint of a class that defines set of abstract methods .

Abstract methods means methods without an implimentations .

And a Class which implements interface must have to provide implimentation of all abstract methods .

We use Interfaces to achieve 100 % abstraction, abstraction means only showing essential data and hiding internal implementing details .

In interface all the methods are by default public abstract.

[ Multiple inheritance supported with the interface. ]

For example : arrayList and LinkedList class both implement List interface which means they must provide implimentation of methods with a same name and same signature that are defined in the List interface such as add() , remove() , size() .

***What is abstract method ?***

***=>*** the method which is not having implimentation , only having declaration part ,such type of method are called as abstract method .

We use abstarct keyword to declare a method as abstract .

For example :

Public abstract void add( );

**What is abstract class ?**

**=>** If the implimentation of class is not complete such type of partially implemented classes are called abstract class .

We use abstract keyword to declare a class as abstract . it says that this class is partially implemented .

And Abstract class contains both abstract method and non abstract methods .

And if class is declared as abstrcat we can not instantiate that class .

And If any class has at least one abstract method then that class must have to declare as abstrct . ]

**String is muttable or immutable?**

**=>**String is immutable means once String object is created , its value cannot be changed.

String are stored in string constant pool. We can create string object by two ways :

1. by using string literals ie. using double quote “ “ for example : String s1 = “hello”
2. And secound using new keyword : for example String s1 = new String(“hello”);

**Why string is immutable ?**

**=>** string is immutable means once string object is created , its value cannot be changed

Because of string immutable it provides security ,means protecting from changing its value .

And string value remains consistence even multiple threads access it .

Overall by making string immutable java ensures that theire value remains consistence , secure and efficient .

**Difference between StringBuffer and StringBuilder ?**

**=>** StringBuffer and StringBuilder classes are used to manupulate string .

StringBuffer :

StringBuffer is synchronized , which means that StringBuffer is thread safe .means multiple threads can not access it simultaneously .

It is slower that StringBuilder.

StringBuilder :

StringBuilder is not synchronized , which means it is not thread safe , means multiple thread can access it simultaneously.

It is faster , because don’t have overhead of synchronization .

**What are generics ?**

**=>** generics provides a way to create classes , interfaces or methods that can work with any data type , rather than specific data type .

**Why we need generics ?**

**=>** consider class below :

class MyList {

private List<String> values;

void add(String value) {

values.add(value);

}

void remove(String value) {

values.remove(value);

}

}

Here we can add only String data type here .

But by using generic we can make it type-safe :

class MyListGeneric<T> {

private List<T> values;

void add(T value) {

values.add(value);

}

void remove(T value) {

values.remove(value);

}

T get(int index) {

return values.get(index);

}

}

**What is static keyword ? explain ?**

**=>** static keyword used with variable or method or block . when we declare variable or method or block as static means they belongs to class itself .

When I declare variable or method as static , it means that there will be only one copy of that variable or methods share for all the instaces of that class .

We can access static variables and methods directly by using class name .

For example : I have static variable count in Test class so I can access It by using Test.count

and similerly with the method suppose I have add() method and which is static means syntax like public static int add() ; so this method I can access by using class name directly

As Test.add();

**Why we use static blocks ?**

**=>** static blocks are used to initialised static variables before any object of class is created .

**Can I write multiple static block ?**

**=>** yes , suppose we have static final variables count1 and count2 and we are intializing it using static block so we can write two static blocks sepratly like

Static {

Count1 = 10;

}

Static {

Count2 = 15;

}

**Can we use final keyword with the static block .**

**=>no we can not use final keyword with static block because**

Final keyword is used to indicate that a variables value cannot be changed once it has been initialised .

And staic block are used to initialized static variables . and static variables themselves can be final .

**What is multiple inheritance ?**

**=>** multiple inheritance is a type of inheritance . in which a class is extends from another two classes .

Java does not support multiple inheritance . because multiple inheritance create ambuiguity issue .

For example we have two classes and they have same method like public void show() in both classes .

And if we create another class which extends these two classes then if create object of extends class and we call show () method . then that object mey get confuse which show() mthod I have to call .

that’s why multiple inheritance is not supported by java .

**Is multiple interface inheritance supported by java ?**

**=>** yes , we can create interface which extends from other two interfaces .

**What is constructor and destructor ?**

**=>** constructor is a special method which is used to intialise member variables of invoking objects .

For example I have class student , in that I have field like name , email , phone no .

So whenever we create object that time we required constructor to intialise value for instance variables .

And in java if we did not provide any constructor in class . compiler provides default constructor .

A destructor is special method in object oriented programming language that is called when an object is to destroyed . the purpose of destructors to release any resource that the object has acuired during its lifetime .

Java does not required destructors . instead java provides garbage collector which automatically manages memory allocation and deallocation of objects .

**Question : java by kiran youtube**

1. **what all data types you know in java ?**

**=> byte int short long**

**Float double**

**Char**

**Boolean**

1. **can you give size required for each dataype ?(in bytes )**

**=> byte - 1 ; short - 2 ; int - 4 ; long - 8**

**Float - 4 ; duble - 8 ;**

**Char- 2**

**Boolean - 1**

1. **what is default value of byte short int long float double boolean char ?**

**=> default value for all byte short int and long is 0**

**Wheras default value for float & double is 0.0**

**And default value for char is blank and default value for boolean is false .**

**[note : String has default value of null ]**

1. **Is String is datatype ?**

**=> No , String is in-built class .which is present in java.lang package**

**It has methods like length() , indexOf() etc .**

1. **can you define a class ?**

**=>**

1. **Can you Define Object ?**

**Composition vs Aggregation**

**=>** HAS-A relationship known as composition or agrregation .

Advantage of HAS-A relationship is code reusability .

**Composition :**

Without existing container object if there is no chance of existing contained object then the relationship between them is called composition.

Ex . CAR has a Engine.

**Aggregation :**

Without existing container object if there is a chance of existing contained object such type of relationship is called aggreagation.

Contained

object

Container object

**Why do we need collections ?**

**=>** Arrays are not dynamic , once an array of particular size is declared . the size cannot be modified . so it create problem to add new element .

Whereas collection is dyamic . they are grow or shink in size as per our need . collections are used to store groups of objects . collections framework provide set of interfaces like list , map , set and classes like Arraylist , linkedlist

, hashmap , treemap , hashset , hashmap .

**What are important method that are declared in collection interface ?**

**=>** add() = to add an element

remove() = to remove an element

size() = return size of collection

isEmpty() = to check collection is empty or not

Clear() = to remove all element

Contains() = to check is object is present in array or not

containsAll() = to check collection present in other collection

addAll() = to add collection of element to another collection

removeAll() = to remove all collection element.

**What is difference between Set and sortedSet interface ?**

**=>** SortedSet interface extends Set interface and Set interface extends Collections interface .

Both Set and SortedSet do not allow duplicates element .

Main difference between set and sortedset is SortedSet interface maintains its element in sorted order . Set interface does not guarenty any order . for example if elements 4 , 5 , 3 are inserted into an implimentation of Set interface . it might store elements in any order .

However if we use SortedSet the elements are sorted . SortedSet implimentation would give output 3 , 4 , 5 .

What is difference between Map and SortedMap ?

=>SortedMap interface extends Map interface and Map interface extends Collections interface .

Both Map and SortedMap store elemnt in key value pairs .

Main difference between Map and sortedMap is SortedMap interface maintains keys in sorted order .

Explain Queue interface ?

=>

**How do you sort Arraylist ?**

**=>** we can sort using Collections.sort( list ); method

**How to convert list to array ?**

**=>** we can covert list to Array by using toArray( arr) method on list . It will return an String Array .

We can use toArray() method on list it will return object array .

Object[] numbers1ObjArray = numbers1.toArray();

System.out.println(Arrays

.toString(numbers1ObjArray));

//[one, two, three, four]

**How to convert Array to list ?**

**=>**  we can convert Array to list using Arrays.asList( arr ) it will return list .

String values[] = { "value1", "value2", "value3" };

List<String> valuesList = Arrays.asList(values);

System.out.println(valuesList);//[value1, value2, value3]

**What is vector class ? how is it different from an arraylist ?**

**=>** Vector has the same operations as an ArrayList. However, all methods in Vector are synchronized. So, we can use Vector if we share a list between two threads and we would want to them synchronized.

What is linkedHashSet ? how it is different from hashset ?

=> LinkedHashSet implements set interface and exposes similar operations to a HashSet. Difference is that LinkedHashSet maintains insertion order. When we iterate a LinkedHashSet, we would get the elements back in the order in which they were inserted.

**What is TreeSet ? how it is different from hashset ?**

**=>** TreeSet implements Set, SortedSet and NavigableSet interfaces.TreeSet is similar to HashSet except that it stores element’s in Sorted Order.

Give example of implementation of navigable map ?

=> TreeMap class implements navigable map interface .

What is priority Queue ?

=>

Difference between peek() method and poll() method in queue ?

=> peek() : peek element gets element with highest prority . (ie minimum number) . it does not remove element .

Poll() : it returns and remove element with highest priority.

Give example of exception handling ?

=>

In what scenario finaly block is not executed ?

=> if we use System.exist() in program . jvm will crash and finaly block will not executed .

Is finally block executed even when there is return statement in the try block ?

=> yes ,

Is try block without catch block is allowed ?

=> yes , try block without catch is allowed , means we hav to provide try block with finaly

But try block without catch and finaly is not allowed.

Explain heirarchy of exception related classes ?

=> throwable class is highest level hirarchy class .

calsses error , exception , and runtimeException extends throwable class .

Class checkedException extends Exception

And

Class uncheckedException extends runtimeExcception

What is difference between an Error and an Exception?

Error is used in situations when there is nothing a programmer can do about an error. Ex: StackOverflowError, OutOfMemoryError. Exception is used when a programmer can handle the exception.

How do you create a Custom Exception Classes?

We can create a custom exception by extending Exception class or RuntimeException class. If we extend Exception class, it will be a checked exception class. If we extend RuntimeException class, then we create an unchecked exception class.

How should the Exception catch blocks be ordered ?

Specific Exception catch blocks should be before the catch block for a Generic Exception.

