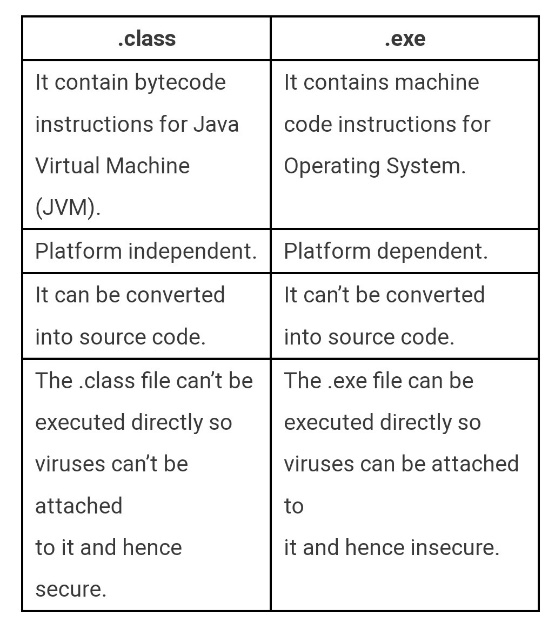
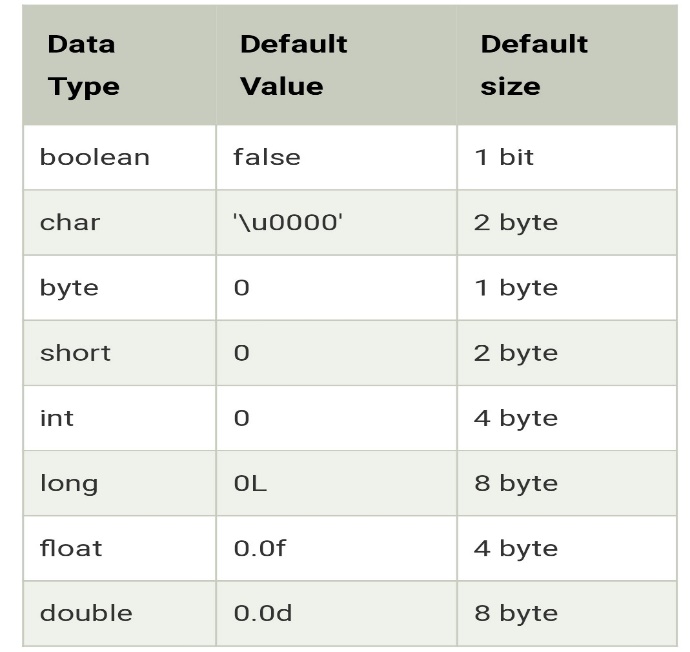
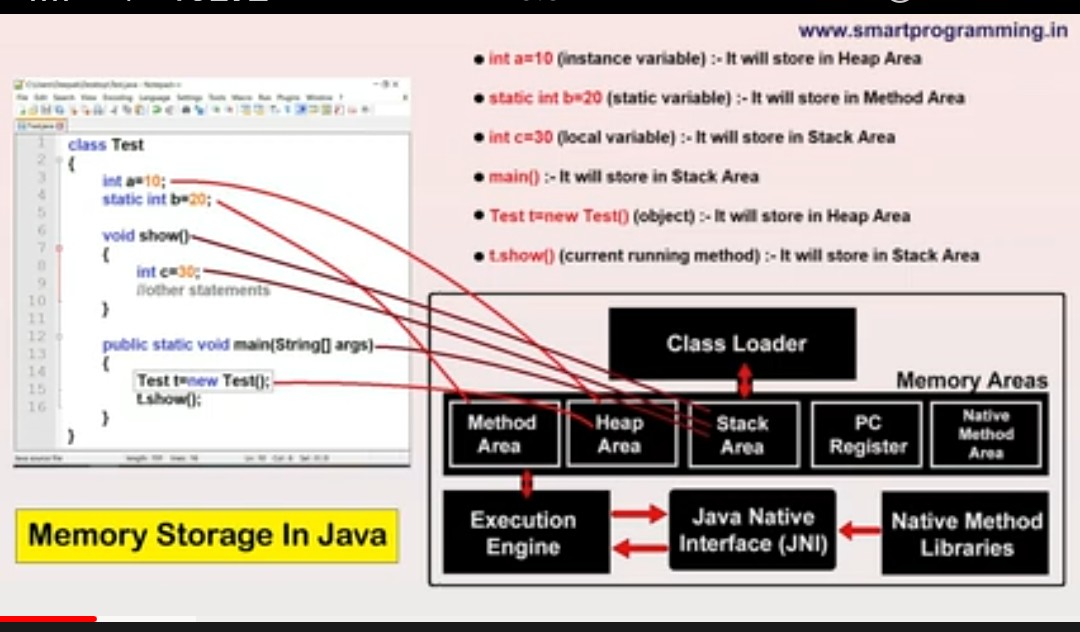
** JAVA interview Q&Ans Q 1) (a)Difference between .class file & .exe file ? (b)what is default value & default size of datatype in java ? **

****

**Q 2).Difference among JDK, JRE,JVM?** JDK stands for Java Development Kit, it contains JRE & java Development Tools, Such as:- java compiler/javac.exe, debugger, java application launcher/java.exe ,java docs, Javaviewer.it is physically present. \*JDK is mainly targeted for a java development, that means we can (a)create java file(with the help of java packages), (b)compile java file(with the help of JavaC) & (c)run a java file(with the help of JVM) **we can say, JDK=JRE + java development tools,**

\*JRE stands for Java Runtime Environment, it contains JVM, java package classes, Runtime Libraries. if we want to run java program we need to have JRE installed in our system.it is physically present **We can say JRE= JVM + java package classes, Runtime libraries.**

\*JVM stands for Java Virtual Machine, it is a software written in c language which executes *compiled java file* or *.class file* containing byte code into machine code. The main tasks of JVM is to load code , verify code, run code,& provide runtime environment libraries.

**Q 3)What is object oriented programming?** Ans:-Object oriented programing is a technique of programing ,in which we create program by using class & object.

**Q 4)Why we need object-oriented programing?** Object-oriented programing helps us to think in terms of real world objects.

**For instance:-** if we are creating a software for Hospital, Hospital has patients so we create a class patients, patient has a name, patient has an address so we create variable using data type for storing name, address etc of patient. Now Hospital has doctors so we create a class doctor. doctor has a name, so we create variable to store name of doctor. Patient is allocated to a doctor who will treat the patient. & so on. So we can see here ,our code is mimicking the real world object.

***Q 5)Features of Java 1.*Platform *Independent.*** Java is a platform independent language because of the bytecode of java. In java when we execute the source code...it generates the .class file comprising the bytecodes. Bytecodes are easily interpreted by JVM which is available with every type of OS (windows, Linux, mac) we install.

***2.Object Oriented****.* Object-oriented programming is a method of programming based on a hierarchy of classes, and well-defined and cooperating objects.

***3. Robust.*** java gives importance to memory management by using the technique called Garbage Collection and Exception handling.

***4. Security*** since java is used on internet, security is an important issue. A security code is asked before a java code is interpreted on internet.

***5. MultiThreading and interactive*** Multithreading means handling more than one job at a time. Java supports Multithreading.

**Q 6)What is class & object?** Class is a user-defined data type which has its own data member & member function, & that can be accessed by creating its own object. That means ,class is a blueprint if we want to use a class we need to create object of the class. Object is an instance of class ,it’s a real world entity.

**Q 7) what are the important pillars of OOP?.** (i)Abstraction:-show only what is necessary.

(ii)Polymorphism:-object acts differently under different circumstances.

(iii)Inheritance:-Parent-child relationship

(iv)Encapsulation:-Hide complexity(hide anything which is complex, which should not be shown outside the object

Q8) what is the difference between Abstraction & Encapsulation? Abstraction means show only what is necessary, For example:-if we have a class employee, the name should be shown outside so we make it public, address has to be shown outside so we make it public

Encapsulation means hide complexity.

Q8. What do u know about java?

Ans:-Java is a high-level ,object-oriented, platform independent , robust, secure programming language. used for application programming like Email, web browsers, gaming software, word processors, graphics software, media player, etc. All of these programs provide an application to the end users, so they are known as application programs. web browsers are used to find information while Gaming software is used to play games. unlike some languages are used for system programming like OS, networking system, website server, data backup server etc.

The system programs are used to programs the operating system software while application programs provide software that is used directly by the users. system programs provide software that are used by other systems such as SaaS application, Computational science applications.

Note:-Maximum Length of string that is possible in java is approx 214 crore or something

What is operating system?

Operating system is an interface between a computer user & computer hardware, it is a software that handles all basic tasks file management , memory management ,process management, handling input & output ,peripheral devices such as disk drives, & printers.

**Collection framework, Collection vs Collections vs collection, Collection properties(Duplicate, Ordered, Null values, Sort)**

**Iterable -> Collection -> (1)List (2)Set (3)Queue These are interface.**

**Collection- Basic operation:** Add, Remove, Search, Retrieve etc.

1. Boolean add(Object O);
2. Boolean addAll(Collection C);
3. Boolean remove();
4. Boolean removeAll();
5. Void clear();
6. Boolean contains(Object O);
7. Boolean containsAll(Collection C);
8. Iterator iterator(); //iterate through the collection
9. Boolean isEmpty();
10. Int size();
11. Object[] toArray();

**(1)List**-> (i) ArrayList (ii) LinkedList (ii) Vector->stack

**(2)Set**-> (i) HashSet (Duplicates are not allowed) (ii) LinkedHashSet (iii) Sorted Set(interface)-> Tree Set(sorted object in asc bydefault)

**(3)Queue**-> (i) Deque(interface)-> Array Queue (ii)Priority Queue

**Map(interface)** -> (i)HashMap->LinkedHashMap (ii)HashTable (iii) Sorted Map(interface) ->Tree Map

**List is used to store collection of elements, List allows duplicate elements.**

**There are 3 subclasses of List-**

**(i)ArrayList-** Internally uses Dynamic Array, Acts like list only, useful for data storage & access.

**Duplicate? - (yes), Ordered? - (index based), Null values?- (yes), Sorted?- (No), Index based?- (yes), Synchronized?- (No)**

**(ii)Vector-** Internally uses Dynamic Array, Acts like an array, useful for thread safe.

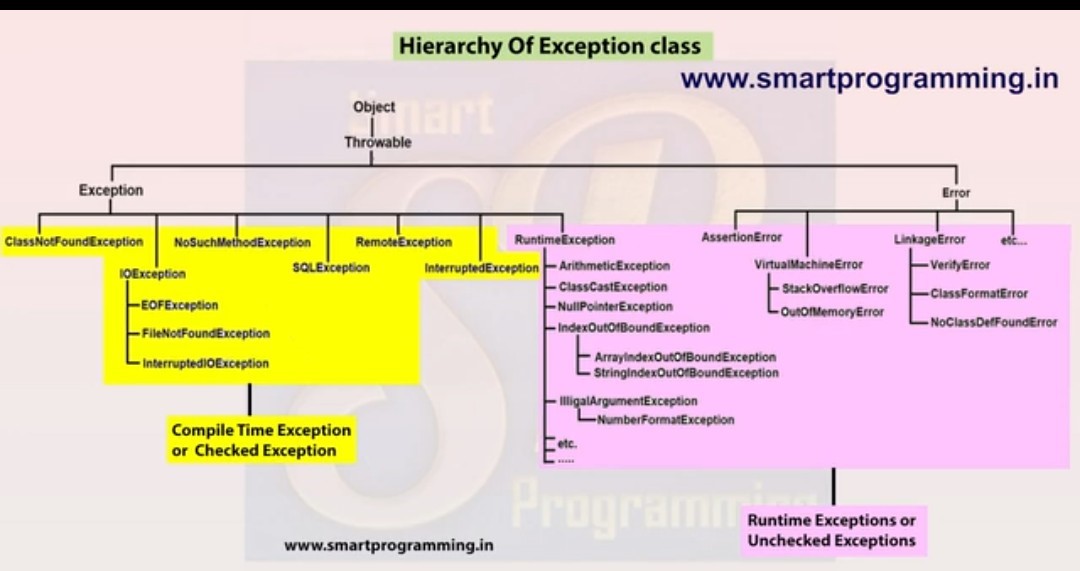
**(iii)LinkedList-** Internally uses Doubly LinkedList, Acts like list & Queue, useful for data manipulation.

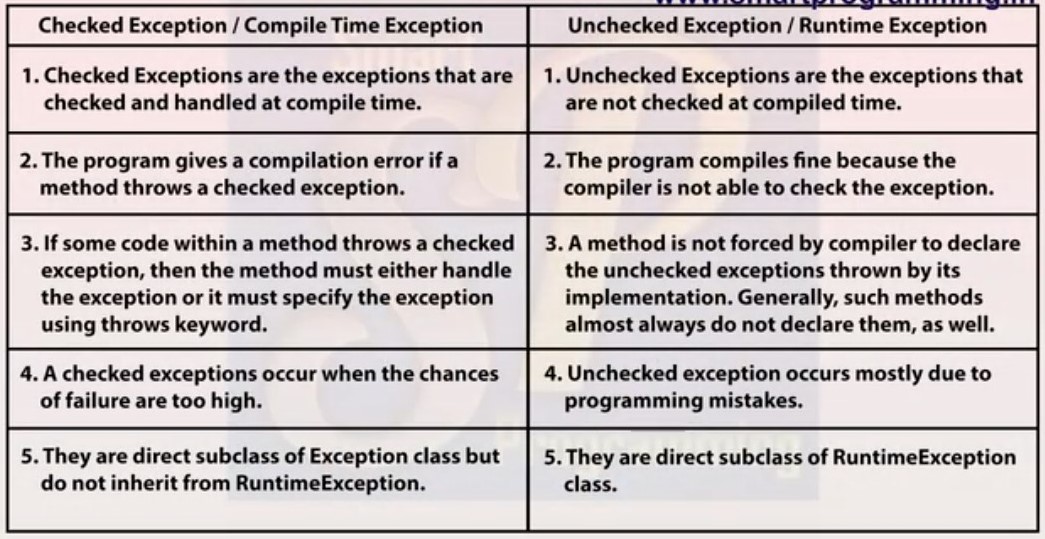
**Extra Methods in LinkedList are-**

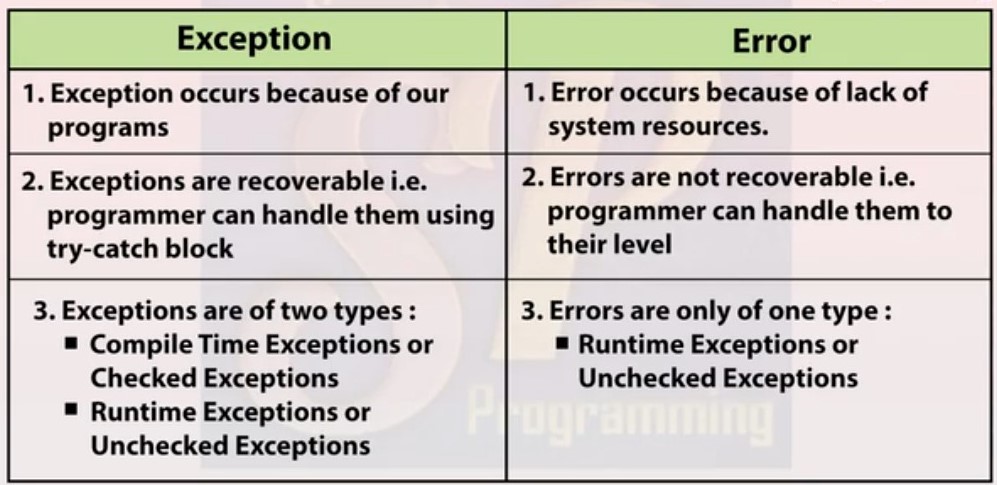
(1)void addFirst(Object o) (2)void addLast(Object o) (3)Object removeFirst() (4) Object removeLast() (5)Object getFirst() (6)Object getLast()

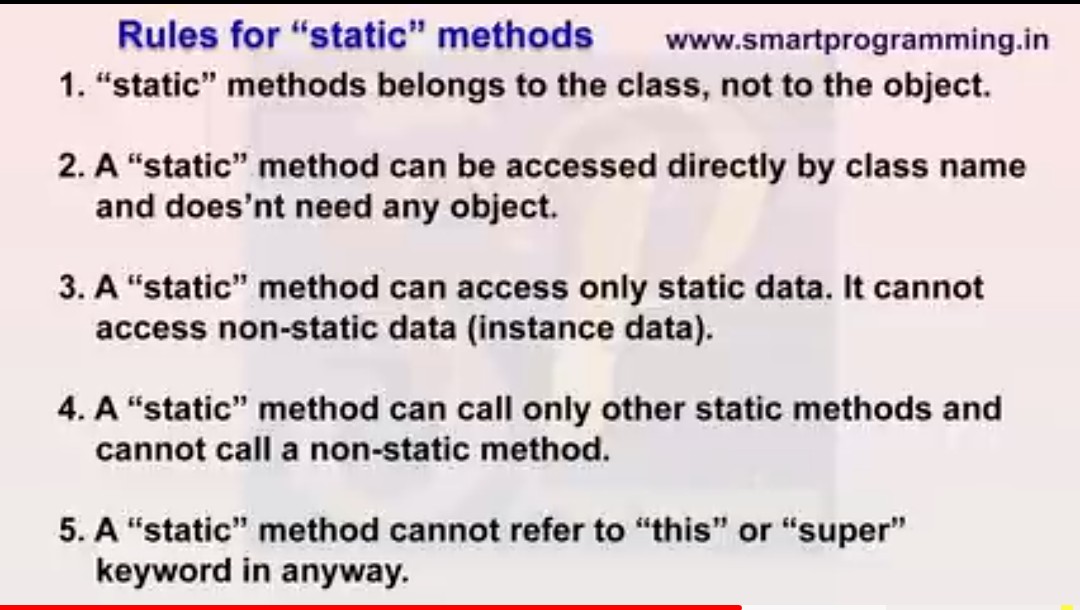
Comparable is an interface, that is available within java.lang package(default package), it is used to order the objects of the user-defined class. it contains only one method which is compareTo(Object o).it provides single sorting sequence only, means you can sort the elements on the basis of single data member only. For ex- it may be name, rollNo, age or anything else.

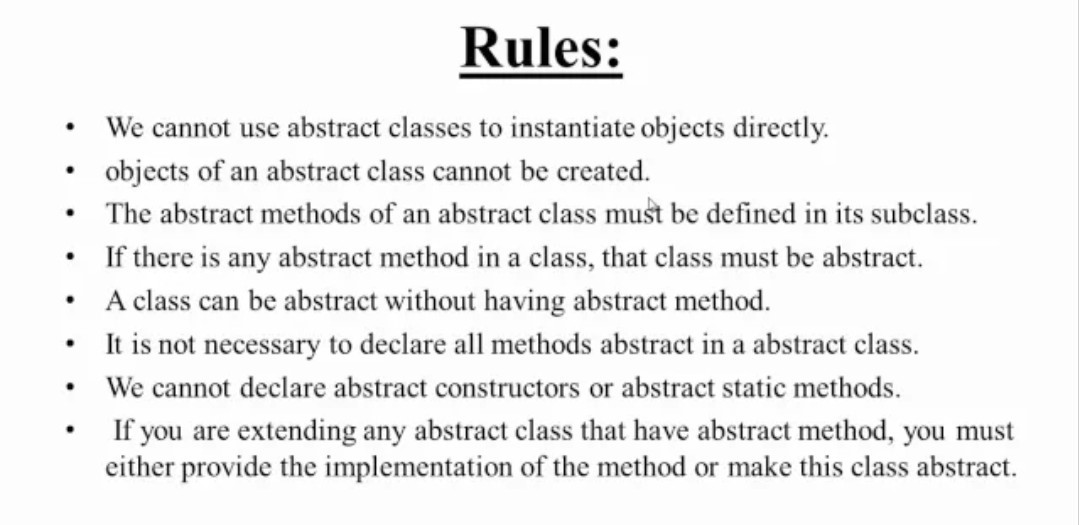
Comparator is an interface, that is available in java.util package, it is used to order the objects of the user-defined class. It contains 2 methods- compare(Object O1, Object2 O2) & equals(Object O1) method. It provides multiple sorting sequences, i.e. we can sort the elements on the basis of any data member, for ex- name, roll no, age, or anything else.

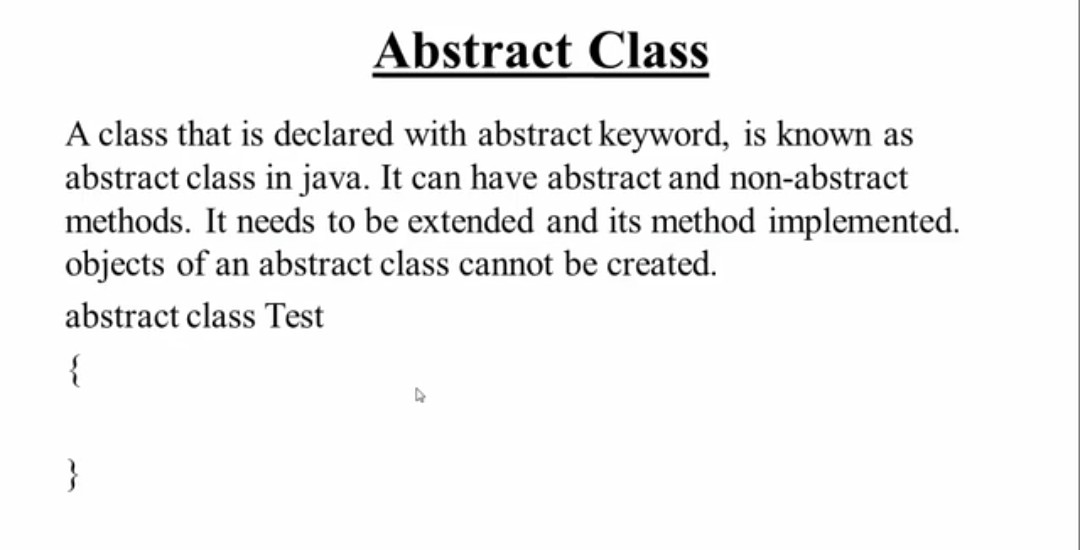


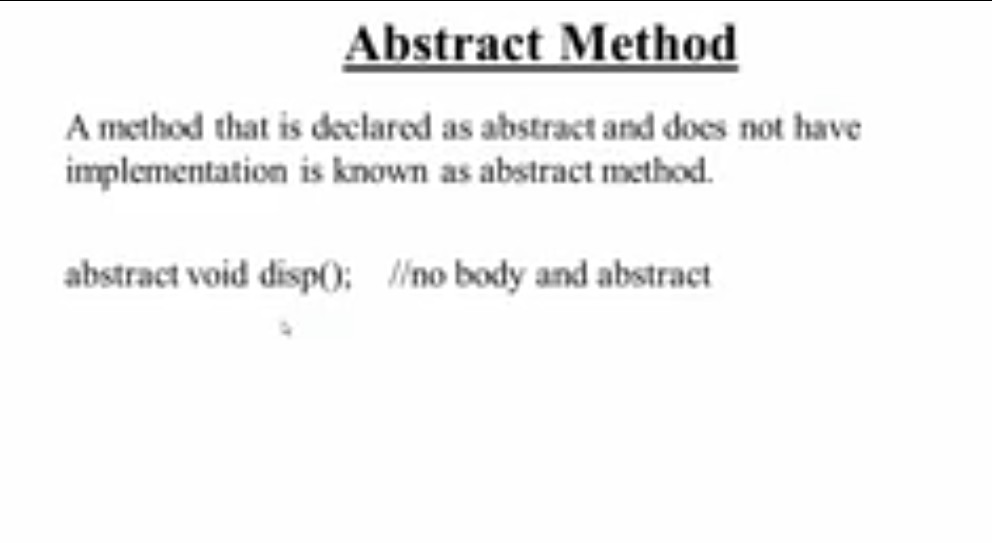












**Permitted Conversion in java**

1. byte to short, int, long, float, double
2. short to int, long, float, double
3. char to int, long, float, double
4. int to long, float, double
5. long to float, double
6. float to double

**(Q).What is difference between HashTable &HashMap?**

**Both classes are used to add key & value pairing, For this we need to add put() method.**

HashTable is synchronized & threadsafe, while HashMap is non-synchronized not therad safe.

In HashTable null value is not allowed while, In HashMap null value is allowed.

(Q1).What is object-oriented programming?

Object Oriented programming is a type of programming that is based on objects rather than just functions & procedures. Individual objects are grouped into classes. OOPs implement real world entities like inheritance, polymorphism, hiding etc into programming. It also allows binding data and code together.

(Q2).Why use OOPs ? OR Advantages of OOPs

Code can be reused through inheritance, thereby reducing redundancy. Data & code are bound together by encapsulation. OOPs allows data hiding therefore, private data is kept confidential(secret). Problems can be divided into different parts making it simple to solve. The concept of polymorphism gives flexibility to the program by allowing the entities to have multiple forms.

(Q3). What are the main features/pillars of OOPs?

(i)Inheritance (ii) Polymorphism (iii) Encapsulation (iv)Data Abstraction

(Q4). Can you call the base class method without creating instance?

Ans:- We can call the base class without instantiating it if- (i) It is static method (ii) The base class is inherited by some other subclass

|  |  |
| --- | --- |
| Object | Class |
| An Object is a real world entity which is an instance of class | A Class is basically a prototype/template/blueprint within which objects can be created. |
| Object acts like a variable of the class. | Binds methods, data together into a single unit. |
| An object is a physical entity. | A class is a logical entity. |
| Objects take memory space when they are created. | A class does not take memory space when created. |
| Objects can be declared when required. | Classes are declared just once. |

(Q5). What is inheritance?

Ans:- Inheritance is a feature of OOPs which allows classes to inherit common properties from other classes. For example- If there is a class such as ‘vehicle’, other classes like ‘bike’, ‘car’ etc can inherit common properties from the vehicle class. This property helps us get rid of redundant code thereby reducing the overall size of the code.

(Q). What is data abstraction?

Data abstraction is a very important feature of OOPs that allows displaying only the important information & hiding the implementation details. For example- while riding a bike, you know that if you raise the accelerator, speed will increase, but you don’t know how it actually happens. This is data abstraction as the implementation details are hidden from the rider.

(Q). How to achieve data abstraction?

Data abstraction can be achieved through:- (i) Abstract class (ii) Abstract method

Before java14, when we created a project, we need to store data so we create object to store the data, & to create object we need to create **class**

that **class** is called POJO(plain old java object) **class** or /DTO(Data transfer object) or entity,let's say object of Student,Employee, etc.

drawbacks- we manually had to create toString,getters etc method only **for** storing the data in our project.

but from java14, we can create record(special type of **class**).

where we can create instance method, **static** method, multiple constructor,

but we don’t need to create getters() method, toString() method, equals(), hashCode() because java auto generates these methods.

but we cannot use setters method to set data, because Record objects are immutable so it has not setters method.

(1)Record is introduced in java14(**if** we create project using below java14 version we cannot use record)

(2)With the help of record we can create special type of **class** to create data carrying object.

(3)Record consists of one or more data fields,java compiler will auto generate getters(), toString(), equals(), hashCode().

(4)we can use multiple constructor

(5)we can create instance method, **static** method.

(6)Records are **final** that means no other **class** **extends** record.

(7)Record objects are immutable once created value cannot be changed.

We can use innjava-17 Records, text-block, switch-expression, sealed classes etc w.

Install plugin “Spring Tool suite” from eclipse marketplace In Eclipse, it shows option springboot, now choose spring starter project,it will open STS window.

Search on google “thymeleaf eclipse plugin” it will show url, copy this url; now go to eclipse->help->add new software->type name: “thymeleaf” [url: paste](url:paste) thymeleaf url

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**What is docker?**

* Docker is an open platform for developing,shipping,running applications.
* Docker is a platform which packages an application & all its dependencies together in the form of containers.

To Use docker: install Docker & wsl(Window substystem for linux)-

(i)Press Win key+R then (ii)type winver & press enter to check windows version, it should be higher 21H2 or higher

Type TaskManager->Performance->Virtualisation:Enabled(by default)

1)Install docker 2)Install wsl by command: “**wsl --install**”

To check wsl on system- “**wsl -l -v**”

1.Check Docker Version - “**Docker -v**”

2.Pull image from docker hub through internet – “**docker pull imageName**”

3.Show all images – type “**docker images**”

4.Search image- “**docker search imageName**”

5.Run image- type “**docker run imageName**”

6.show all docker container- “**docker ps -a**”

7.open container in detach mode & give name- “**docker run --name containerName -d imageId**”

8.open container in detach mode & give name- “**docker run --name containerName -d imageName**”

9.Run image forever- ”**docker run --name openjdk2 -it -d openjdk**”

10.show all running image container- “**docker ps**”

11.To enter to the container- “**docker exec -it containerId commandName**”

12.To find details of the container- ”**docker inspect containerId**”

13. to leave container- “**/exit**”

In case of javaContainer these commands can be executed we can create class as well-

jshell> System.out.println("hello india")

**Q)Some Commands to use mysql database?**

i)Pull image- **“Docker pull mysql”**

ii)Run Mysql first time- **“docker run mysql”(it will set any mysql name automatically)**

**“docker run --name mysqlDbName -e MYSQL\_ROOT\_PASSWORD=root -d mysql”**

ii)Find Details- “**docker inspect mysqlDb**”

**we need to give these commands every time when restart mysql:-**

**(iii)start mysql: ”docker start mysqlDb”**

iii)To go inside mysqlDb: ”**docker exec -it mysqlDb bash**”

iv)To insert mysql password: ”**mysql -p**”

(vi)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q)Some Commands to use nginx server?**

(i)Pull nginx image: “**Docker pull nginx**”

(ii)Run nginx server: “**docker run --name nginxServer -d -p 8080:80 nginx**”

Now we can access container on browser- localhost:8080

**Q)Some Commands** **to use apache server?**

(i)Pull apache image: “**Docker pull httpd**”

(ii) Run apache server: “**docker run --name httpdServer -d -p 8081:80 httpd**”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Some general commands for all containers:-**

**(i)**Run image- type “**docker run imageName**”

**(ii)** Restart any container**:** ”**docker start containerName,containerId,containerStarting3digits**”

**(iii)**Stop any container: “**docker stop containerName, containerId, ContainerStarting3digits**”

**(iv)**Remove conatiners: “**docker rm conatinerId1 conatiner2…”**

**(v)**Remove images: “**docker rmi imageName**”

**Some other commands-**

**“docker login”**

**“docker commit”**

**“docker push”**

**“docker copy”**

**“docker logs”**

**If we want docker Container store our data: run “docker volume:”**

**“docker logout”**

**Q)How to create our own image?**

**(i)Create a folder (ii)open in vs code (iii)create a file with name Dockerfile**

**(iv)Write these commands in the Dockerfile-**

#Base image for ubuntu

FROM ubuntu

#Author name

MAINTAINER pragya

#Update command

RUN apt update

#Start up executable

CMD ["echo","this is my first docker image"]

#For python-

FROM python

WORKDIR /use/src/myapp

COPY . /usr/src/myapp/

CMD [“python3”,”main.py”]

**For java-**

FROM openjdk

WORKDIR /usr/src/myapp

COPY . /usr/src/myapp/

RUN javac DockerTest.java

CMD ["java", "DockerTest"]

**For SpringBoot-**

FROM openjdk

WORKDIR /usr/src/myapp

COPY . /usr/src/myapp/

CMD ["java", “-jar”, "jarprojectname.jar"]

EXPOSE 9090

**(v)For creating image- ”docker build -t imageName .”**

**(vi)For creating container- “docker run --name projectName -it -p 9090:9092 -d imageName”**

**To use Docker-Compose tool-**

**(i)Create a folder “docker-compose” in which create a file “docker-compose.yml”, write this configuration in docker-compose.yml**

**(ii)Again create a folder in docker-compose in which we add jar file of our sboot-app & create Dockerfile**

**In Dockerfile-**

FROM openjdk

WORKDIR /usr/src/bootapp

COPY . /usr/src/bootapp/

CMD ["java","-jar",blog-app-apis-0.0.1-SNAPSHOT.jar]

**In docker-compose.yml-**

# creating 2 services - (i)database service & (ii)app service

# create network to communicate each other

# To run this cmd - "docker-compose up -d" ,where d stands for detach mode means app will be running in the background.

# now hit this url on browser -> localhost:8085

version: '3'

services:

#1) db service

  dbservice:

    image: mysql

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

      # - MYSQL\_USERNAME=pragya

      #

    volumes:

      - ./data:/var/lib/mysql

    ports:

      - :3306

    networks:

      - bootapp

#2) app service

  appservice:

    build: ./sboot-blogapp

    depends\_on:

      - dbservice

    environment:

      - spring.datasource.url=jdbc:mysql://dbservice:3306/blog\_app?createDatabaseIfNotExist=true

      - spring.datasource.username=root

      - spring.datasource.password=root

    ports:

      - 8085:9292

    networks:

      - bootapp

networks:

  bootapp:

**don’t need to give these command-**

**For creating image- ”docker build -t imageName .”**

**For creating container- “docker run --name projectName -it -p 9090:9092 -d imageName”**

**we need to give only 1 command in docker-compose- “docker-compose up -d”**

**C:\Users\psinh\Desktop\docker\docker-compose>docker-compose up -d**

**\*Dockerizing React with nginx server for production**

**Write in docker file**

# node block

FROM node:alpine3.16 as nodework

WORKDIR /reactapp

COPY package.json .

RUN npm install

COPY . .

RUN npm run build

# nginx block

FROM nginx:1.23-alpine

WORKDIR /usr/share/nginx/html

RUN rm -rf ./\*

COPY --from=nodework /reactapp/build .

ENTRYPOINT ["nginx", "-g", "daemon off;" ]

**Write in Docker-compose.yml**

version: "3"

services:

  web:

    container\_name: dockercompose-reactContaineron-nginx

    build: .

    ports:

      - 3000:80

**Create file – .dockerignore to ignore some files(README.md, node\_modules, gitignore, Dockerfile,.git, .dockerignore, .env)**

**Run “docker build -f Dockerfile.dev -t reactapp(anyName) .” to build image**

**Run image to build container- “docker run -it -d –name reactApp -p 3000:3000 reactapp(builded imageName)”**

**Serving Image After building with the help of nginx server**

**(Q).What is maven?**

Ans:- Maven is a project management tool which helps to bring jar from internet to our project.

**(Q).Difference between Core java & Advance java?**

Ans:-There is nothing like to differenciate between core java & advance java. All the concepts we use in core java are used in advance java.

The application of core java is basically the advance java. Advance java is not a 1% without core java.

**(Q).What is JDBC?**

JDBC stands for JAVA DATABASE CONNECTIVITY. It is a java api. It has some classes & interfaces used to interect with database. JDBC allows java program to execute sql statement & to manage database connections.

**(Q)What is compile time error?**

Compile time error means java compiler identifies the syntax error at the time of compilation. & without compilation java does not create .class file. That means we have to compile program which should be error free & then compiler creates .class file of the program & then we can run the program. **Some example of compile time errors-** Missing braces, Missing semicolon, Missing double quote in string, use assignment operator(=) instead of equal to operator(==).

**(Q)What is Runtime error?**

Several time program may compile successfully & compiler creates the .class file of the program. But when it comes to running the program, it shows error called runtime error. **Some example of Runtime errors-** Divide by zero, Access the element that is out of bound array, Passing the parameters with invalid range.

**(Q).What is Exception?**

Exception is nothing but an unexpected,unwanted,abnormal condition/event which occurs during the execution of program.That terminates/interrupts normal flow of execution of the program.That is called exception.

There are 5 keyword in java which are used for exception-handling:- try-catch, throw, throws, finally

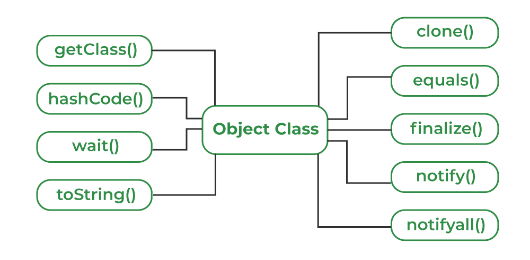
**(Q).Which are the types of exceptions are there in java?**

There are 2 types of exceptions:- (1)Compile time(checked exception) (2)Runtime (unchecked exception).

The **throws** statement is used by a method to specify the types of exceptions the method throws. If a method is capable of raising an exception that it does not handle, the method must specify that the exception have to be handled by calling a method. This is done using throws statement, The throws clause lists the type of exception that a method might throw exception.

**(Q)What is Object class in java?**

Object class is super class of all the classes in java. It is present in java.lang package(default package).Every class in java is directly or indirectly derived from Object class. if a class doesn’t extend any other class then it is direct child of Object. If extends another class then it is indirectly derived.



public int hashCode(), public String toString(), public Class getClass(), public Object clone() throws CloneNotSupportedException, public boolean equals(Object obj), public final void wait() throws InterruptedException, public final void wait(long timeout)throws InterruptedException, public final void wait(long timeout,int nanos)throws InterruptedException public final void notify(), public final void notifyAll(), protected void finalize() throws Throwable

**IMPORTANT POINTS OF JAVA:**

Here are some important points about Java:

1. Java is a high-level, object-oriented programming language that was first released in 1995.
2. Java is platform-independent, which means that code written in Java can run on any platform that has a Java Virtual Machine (JVM) installed.
3. Java code is compiled into bytecode, which can then be executed by the JVM.
4. Java is known for its “write once, run anywhere” philosophy, which makes it a popular choice for cross-platform development.
5. Java provides automatic memory management through garbage collection, which makes it easier to write and maintain code.
6. Java has a vast standard library that provides a wide range of tools for common programming tasks.
7. Java is widely used in enterprise applications, web development, and Android app development.
8. Java is a strongly typed language, which means that every variable and expression has a specific type that must be declared before use.
9. Java has a robust exception-handling mechanism that makes it easier to handle errors and unexpected behavior in code.
10. Java supports multithreading, which makes it possible to write programs that can perform multiple tasks simultaneously.

**Advantages of Java:**

1. Platform independent: Java code can run on any platform that has a Java Virtual Machine (JVM) installed, which means that applications can be written once and run on any device.
2. Object-Oriented: Java is an object-oriented programming language, which means that it follows the principles of encapsulation, inheritance, and polymorphism.
3. Security: Java has built-in security features that make it a secure platform for developing applications, such as automatic memory management and type checking.
4. Large community: Java has a large and active community of developers, which means that there is a lot of support available for learning and using the language.
5. Enterprise-level applications: Java is widely used for developing enterprise-level applications, such as web applications, e-commerce systems, and database systems.

**Disadvantages of Java:**

1. Performance: Java can be slower compared to other programming languages, such as C++, due to its use of a virtual machine and automatic memory management.
2. Memory management: Java’s automatic memory management can lead to slower performance and increased memory usage, which can be a drawback for some applications.

1.)Basic – Class,Object, DataTypes, TypeCasting, Arrays, String, Loops

2.)Intermediate- Oops Concept (Inheritence,Polymorphism),FileHandling, MultiThreading, Collection Framework ,Networking etc .

3.)Advance- Database Connectivity, GUI, JDBC, Spring Servlet, ORM, Spring MVC, SpringBoot etc

4.)Practical Experience(3-4 live project)

5.)Additional- Git,Cloud, Docke, CI/CD Pipeline etc. 6.)Problem Solving concept 7.)Soft-skills

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**(Q)What is springboot?**

Springboot is a java framework which makes it easier to create & run java application.

Springboot is a module of spring framework which facilitates Rapid Application Development(RAD) capabilities.

Springboot simplifies the configuration & setup process, allowing developers to focus more on writing code for their application.

Springboot solves many developers problems-

(i)configuration (ii)Dependency management (iii)Embedded server& so on..

**(Q)Why springboot over spring?**

Springboot provides many advantages over spring framework-(a). Easy to use- Remove boilerplate codes

**(Q)Working of springboot-**

**(Q)How springboot starts?**

(i)It starts by calling main() method of main class. (ii)The run() method of SpringApplication is called.This method starts the application by creating an ApplicationContext & initializing it. (ii)Once the ApplicationContext is initialized, the run() method starts the application’s embedded web server.

By default it scans current packages using @SpringBootApplication to scan another package use @ComponentScan(“packageName”)

**(Q)Top Springboot annotations-**

**(Q)What are the springboot starters?**

Starters are a collection of pre-configured dependencies that make it easier to develop particular kinds of applications.

These starters include all of the dependencies,version control & configuration needed to make certain features of a Spring boot application functional.

**(Q)What are the key dependencies of springboot?**

**(Q)What is spring-boot-starter-parent?**

Spring boot starter parent is a starter project that provides the default configuration for spring based applications. Dependency management feature manages the versions of common dependencies. Provide the default compiler level as java1.8 & UTF-8 source coding. Provide a default configurations for maven-plugin such as- maven-surefire-plugin,maven-jar-plugin,maven-failsafe-plugin.

**(Q)Can we use only springboot dependency feature & configure maven plugin manually?**

(i)Yes (ii)We don’t inherit from the spring-boot-starter-parent pom. (iii)Include the spring-boot-dependencies inside dependencyManagement section as an import scope.

**(Q)What is springboot CLI & what are its benefits?**

Download springboot-cli -> open a folder using cmd -> type “set path=springboot-cli\_path until bin”

Create project using command: “spring init -d=web,data-jpa project-name“

Some commands : “spring version”,“Spring init –list”

**(Q)What is thymeleaf?**

Thymeleaf is a java-based server-side templating engine used in java web applications to render java web pages.

**(Q).What is inversion of control (IOC)?**

Inverting the control of creating object using new keyword to container or framework.

**(Q).Explain Spring Bean-LifeCycle?**

In spring, a simple java object is called bean .

LifeCycle means- (i)How object is created (ii)How object behaves(in whole life) (iii)How object destroies.

**Spring bean life-cycle is maintained by IOC container-** (i)Container gets started. (ii)Container creates bean as per request. (ii)dependency is created. (iv)dependency is injected. (v) Bean is destroyed when container closed.

**There are 3 ways to implement bean life-cycle -** (i)Using annotations (@PostConstruct, @PreDestroy) (ii)Using java interfaces (IniializingBean, DisposableBean)(ii)Using xml

**@PostConstruct-** PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization.

**(Q).What is Bean Factory? What are the difference between BeanFactory & ApplicationContext ?**

Bean Factory is a top level interface which provides some features to manage spring bean, & it extends ApplicationContext interface which provides some extra features.

**BeanFactory- (i)**Bean instantiation/wiring

**ApplicationContext- (i)**Bean instantiation/wiring  **(ii)**Automatic BeanPostProcessor registration Automatic **(iii)**BeanFactoryPostProcessor registration **(iv)**Convenient MessageSource access(for i18n) **(v)**ApplicationEvent publication

**(Q)Difference between setter & constructor injection in spring?**

In Constructor injection, it is important to remember the type & order of constructor parameters. Constructor injection is for mandatory dependencies, ex- If we want to create a phone that depends on battery then use constructor& setter is for optional ex- if we want to create a phone that doesn’t depend on battery then use setter..

We can use java configuration in place of xml in spring

**(Q)What are the different modules in spring?**

The Core container modue Application context module AOP module JDBC abstraction & DAO module

ORM module(Object/Relational) 6 Web Module 7Test

**(Q)What is the difference between @Autowired & @Inject annotation in spring?**

@Inject annotation also serves the same purpose as autowired

The main difference between them is that @Inject is a standard annotation for dependency injection & @Autowired is spring specific

**(Q)Difference between @Bean & @Component?**

@Component is used on the class, @Bean is used on the method;

We can write our own login if we create bean, in component we can not write our own code

**(Q)What is autowiring in spring? What are the autowiring modes?**

Injecting the beans automatically, we don’t need to write expicit injection logic.

(i)no- this is default mode, it means autowiring is not enabled.

(ii)byName- injects the bean based on the property name, it uses setter method

(iii)byType- injects the bean based on the property type, it uses setter method.

(iv)constructor- it injects the bean using constructor.

**(Q)What are the different bean scopes in spring?**

(i)Singleton-The bean instance will be created once & the same instance will be returned by the IOC container, it is default scope.

(ii)Prototype-The bean instance will be created each time when requested.

(iii)Request-the bean instance will be created per Http Request

(iv)Session -The bean instance will be created per http session.

(v)Globalsession- The bean instance will be created per http globalsession.it can be used in portlet context only.

**(Q)How to use whisper to generate subtitle of a video?**

**Whisper is an open-source AI-powered speech recoginition developed by OpenAi.**

**Create a folder, in that folder create a Dockerfile & data folder(keep your video here).**

**Write these in Dockerfile-**

FROM ubuntu

RUN apt update && apt install ffmpeg -y

RUN apt install python3 -y && apt install python3-pip -y

RUN pip3 install -U openai-whisper

RUN pip3 install setuptools-rust

ENTRYPOINT [ "bash" ]

**Now run one by one the following commands-**

(i)To build image: “**docker build -t imageName .**” ((.)means Dockerfile is in current folder)

(ii)To build container: “**docker run -it --name containerName -v ${pwd}:/data imageName**”

(iii)Generate Subtitle using whisper- “**whisper videoName.mp4 --language Hindi --task translate**”

**For creating file if needed-**

Install vim- “**apt install vim**”

Create file using vim- “**vim abc.txt**”

**Relation Mapping-** (i)OneToOne – Ex- One user has one Aadhar card & One Aadhar card has one user.

(ii)OneToMany – Ex-One post has multiple likes.

(iii)ManyToOne- Ex- One post is posted by one user.

(iv)ManyToMany- Ex-Multiple users can save multiple posts.

**Spring Security-** Impelment Spring Security in our application for securing our endpoints. We can use Login-Register functionality.

As we add this dependency in pom.xml, all api/urls are secured, To access those urls It gives username:user password:on console .

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

Implement Register & Login method

**Reels Api- (i)Create Reels (i)Get all Reels (iii)Get user reels**