

JAVA

1) WHAT IS JDK ?

JDK (Java Development Kit) :-

It is a software development environment used for developing java program/applications.
It includes → a) JRE (Java RunTime Environment)
b) JVM (Java virtual Machine)

2) WHAT IS THE LATEST VERSION OF JAVA ?

JDK 1.11

3) EXPLAIN ADVANTAGE OF JAVA ?

i) Easy to Learn

Easy to use & is therefore easy to write, compile, debug & learn than other PL.

ii) It is Object-Oriented

iii) It is platform independent

iv) It is secure

v) It is portable.

vi) It is robust.

4) EXPLAIN JAVAC & JAVA COMMANDS ? IN WHICH FOLDER THEY ARE AVAILABLE ?

JAVAC = To compile the program

JAVA = To launch the program

In bin folder they are available.

5) WHAT IS THE SIGNATURE OF MAIN METHOD ?

(Methodname (Arguments)) ;

6) EXPLAIN KEYWORDS ? LIST SOME ?

Keywords :-

i) They are the predefined reserved word.

ii) Each keyword has some predefined meaning.

iii) There are 53 keywords in which 3 are literally

some → ~~float~~ float ~~goto~~ goto ~~this~~ this
~~Abstract~~ abstract ~~double~~ double ~~while~~ while ~~try~~ try
~~Byte~~ byte ~~char~~ char ~~final~~ final ~~catch~~ catch
~~short~~ short ~~boolean~~ boolean ~~finally~~ finally ~~private~~ private
~~Int~~ int ~~for~~ for ~~else~~ else ~~default~~ default
~~long~~ long ~~if~~ if ~~super~~ super ~~protected~~ protected

7) EXPLAIN IDENTIFIERS OF JAVA ?

Identifier :-

i) They are user defined words & alphanumeric.

ii) These are used as method name, class name, Var. name,

iii) It shouldn't start with numbers

iv) A space shouldn't ^{be used} start ⁱⁿ an identifier.

v) Keywords shouldn't be used as identifier.

vi) It can consist of only 2 symbols i.e ' \$ & '.

8) EXPLAIN ABOUT LITERALS ?

Usage of data by not storing is called as literals. (Technical name of value)

3 types of literals,

i) Number

ii) Character

iii) Boolean

9) EXPLAIN WHY JAVA IS PLATFORM INDEPENDENT.

Java is platform independent because the .class file which is generated after successful compilation can be executed in any other OS like mac, windows etc.

10) EXPLAIN THE DIFFERENCE BETWEEN COMPILER & INTERPRETER?

COMPILER

It will check the error of code & convert the code into .class file (i.e. in bytecode).

INTERPRETER

It can convert any line of code in O & I.

11) IS JVM IS PLATFORM INDEPENDENT?

No.

12) WHAT IS RULE & CONVENTION?

RULE :-

It is mandatory to follow.

Convention :-

It is recommended to follow

13) EXPLAIN ABOUT JRE?

It is a platform where JVM & OS together convert bytecode of .class file into machine level language.

15) EXPLAIN VARIABLES & ITS TYPES?

Variables :-

- 1) Variables are the memory locations used for storing the data.
- 2) Types of variable →
 - 1) Static variables
 - 2) Non-static variables
 - 3) Local variables

16) MENTION ALL THE PRIMITIVE DATATYPES IN JAVA?

Byte Int Float Char
Short Long Double Boolean

17) EXPLAIN ABOUT PRIMITIVE VARIABLES?

Primitive Variable :-

It is used to store only primitive data (i.e. numbers, characters, Boolean)

18) EXPLAIN REFERENCE VARIABLE?

Reference variable :-

It is used to store the non-primitive data.

19) WHAT IS STRING IN JAVA?

String :-

It is collection data with enclosed within double quotes.

20) EXPLAIN METHOD IN JAVA? USES?

Method :-

It is a set of statements which is used for performing any tasks.

- Uses :-
 - 1) Code reusability
 - 2) Reduce no. of lines of code.

21) WHAT IS THE GENERAL SYNTAX OF METHOD?

Access Specifier	Access Modifier	Return Type	Method (Arguments)	Name
------------------	-----------------	-------------	--------------------	------

22) WHAT IS THE DEFAULT EXECUTION METHOD?

Main method.

23) CAN WE COMPILE JAVA PROGRAM W/O MAIN METHOD?

Yes.

24) CAN WE EXECUTE JAVA PROGRAM W/O MAIN METHOD NO.

Because JVM calls the main method for execute.

Ans

Q&A



29) DIFFERENCE BETWEEN STATIC & NON-STATIC MEMBERS ?

25) EXPLAIN OBJECT ?

Object :-

Object is an entity which has its own state & behaviour.

26) EXPLAIN CLASS ?

Class :-

Class is a blueprint or a defn block which is used to describe an object.

27) EXPLAIN THE MEMBERS OF CLASS ?

The things which is declared within the scope of class is called as members of class.

28) EXPLAIN STATIC & NON-STATIC MEMBERS ?

Static Members :-

It is declared within class body & outside methods.

It is declared using static keywords.

It is stored in a single copy in the memory.

It is loaded in static pool Area.

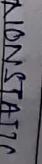
Non-static Members :-

It is declared within class body & outside methods.

It is declared without static key words.

It is stored in multiple copies in the memory.

It is loaded in heap area.



30) DIFFERENCE BETWEEN STATIC & NON-STATIC MEMBERS ?

i) declared using static is declared w/o static keywords.

ii) stored in a single copy w/o storing multiple copy in the memory.

iii) loaded in static pool area w/o loaded in heap area.

iv) classname.membername w/o Objectname.membername

v) EXPLAIN GLOBAL VARIABLE ?

Global Variable :-

It is the one which we can access in all parts of program.

31) EXPLAIN HOW CAN YOU ACCESS STATIC & NON-STATIC GLOBAL VARIABLE ?

Static global variable →

classname.variableName;

Non static global variable →

new classname().variableName;

32) WHAT WILL BE THE DEFAULT VALUE OF GLOBAL, REFERENCE & BOOLEAN FA VARIABLE ?

Global variable → 0

Reference variable → null

Boolean variable → false

34) DIFFERENCE BETWEEN LOCAL & GLOBAL VARIABLE

GLOBAL

It is used in all parts. It is used only in that of program directly. It is available directly.

It is stored in specific area according to their types.

It is declared inside a function.

It is stored in stack.

It is stored in stack.

It is declared outside any function.

It is stored in stack.

It is declared directly in specific method.

It is available directly.

It is declared directly.

It is declared directly.

It is declared directly.

It is declared directly.

NON-STATIC BLOCK :-

It is used to initialise both static & non-static members.

It is declared using static keyword.

It is executed at the time of object creation.

It is executed at the time of object creation.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is executed at the time of class loading but only once in a execution cycle.

It is declared directly.



4.2) EXPLAIN ABOUT CONSTRUCTORS & DIFFERENT TYPES?

Constructors :-

- i) It is a special member of class which is used to initialize non-static variables/fields.
- ii) It is not stored in the memory.
- iii) Every class in java must have a constructor.

& types →

- a) Argumented constructors.
- b) Non-argumented / zeroarg constructor

4.3) WHAT IS DEFAULT CONSTRUCTOR?

When the programmer doesn't give the constructor then compiler by default gives a constructor to the class, that is called as default constructor.

4.4) WHAT IS CONSTRUCTOR OVERLOADING? DIES?

Constructor Overloading :-
Constructor having same name but different with same arguments like by datatype, sequence or length.

Use :-
We can give different input to a particular class.

- i) EXPLAIN NON-STATIC VARIABLE HIDING?
- ii) If local variable has same name as of real instance variable then local variable hide the instance variable & is called as non-static variable hiding.
To achieve that instance variable we use this keyword.

4.5) WHAT IS THIS CALLING STATEMENTS? USES

This statement is used to call the code of same class

- i) It has to be written explicitly.
- ii) It is written in 1st statement of constructor.

4.6) WHAT IS RECURSION?

- i) A func. calling itself is called recursion.
- ii) It is not allowed in constructors.

4.7) RECURSION WHILE CONS. OVERLOADING WILL RESULT IN COMPILE TIME OR RUN TIME ERROR

Compile Time error

4.8) WHAT IS 'THIS' KEYWORD?

- i) This keyword is used to holds the address of current object.
- ii) It is the & can make the difference b/w local var. & global var.

- i) EXPLAIN NON-STATIC VARIABLE HIDING?
- ii) If local variable has same name as of real instance variable then local variable hide the instance variable & is called as non-static variable hiding.
To achieve that instance variable we use this keyword.

50) USE OF THIS KEYWORD?

It holds the address of current object.

It is used to make the difference between local variable & non-static global variable of same name.

51) EXPLAIN HAS-A RELATIONSHIP?

One class having an object of another class & its data members in called as has-a relationship.

52) EXPLAIN INHERITANCE & DIFFERENT TYPES OF INHERITANCE?

Inheritance is a copy principle which is used to acquire the properties of 1 class to another class.

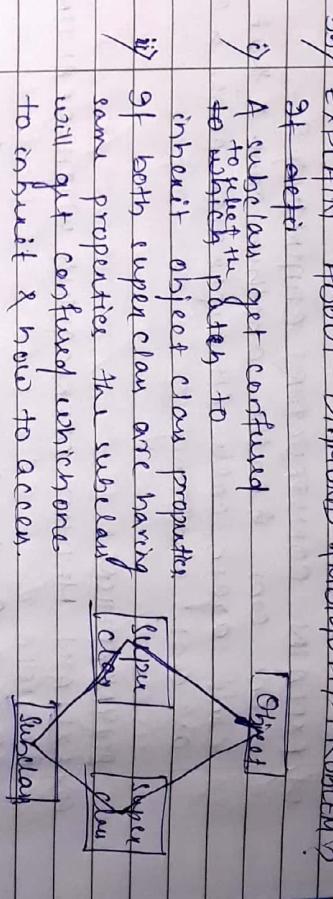
5 types →
1) Single level inheritance

2) Multi level inheritance

3) Hierarchical inheritance

4) Multiple inheritance

5) Hybrid inheritance



54) EXPLAIN ABOUT DIAMOND AMBIGUITY PROBLEM?

It refers

i) A sub class get confused to which parent to inherit object class properties

ii) If both super class are having same properties the sub class will get confused which ones to inherit & how to access.



55) DOES JAVA SUPPORTS MULTIPLE INHERITANCE?

Yes, java supports multiple inheritance.

Multiple inheritance doesn't support in class body but supports in interface.

56) USE OF EXTENDS KEYWORD?

It is used to extend the properties of 1 class to another class or interface.

1 interface class to another interface class.

5.8) DIFFERENCE BETWEEN THIS & SUPER KEYWORD,

This ^{keeps super} holds the object ⁱⁿ It holds the object of current class. of superclass.

5.9) EXPLAIN CONSTRUCTOR CHAINING?

One cons. & calls another cons. & that cons. again calls its superclass cons. is called as cons. chaining.

POLYMORPHISM:-

6.0) EXPLAIN METHOD OVERLOADING WITH REAL TIME EXAMPLE?

Method Overloading:-

Method having same name but different with their args. is called as method overloading.

Ex. class loginUsingData {
 public void login (long phnum, String pwd)
 {
 SOP ("Login using phone & pwd");
 }

 public void login (int a, String pwd)
 {
 SOP ("Login using pin & pwd");
 }
 public void login (String Face)
 {
 SOP ("Login using face recognition");
 }
 }

6.1) EXPLAIN METHOD OVERRIDING WITH REAL TIME EXAMPLE?

Method Overriding:-

Method with same declaration but different with their implementation is called as method overriding.

Ex. class Gmail {
 void compose () {
 SOP ("full screen");
 }

 class Gmail2 extends Gmail {
 void compose () {
 SOP ("pop up");
 }
 class M {
 SOP ("pop up");
 args
 gmail ref. new Gmail2();
 ref. compose();
 }
 }

class FaceBook { public void login (String[] args) { class LoginUsingData { SOP ("Login using Face"); args gmail ref. new LoginUsingData(); ref. compose(); } }

 class LoginUsingData {
 SOP ("Login using Face");
 args
 gmail ref. new LoginUsingData();
 ref. compose();
 }

 class Gmail {
 SOP ("Login using Face");
 args
 gmail ref. new Gmail();
 ref. compose();
 }

5.10) EXPLAIN POLYMORPHISM?

When we want to give multiple data of different types to a method then we go for method overriding.

5.11) EXPLAIN POLYMORPHISM?

5.12) EXPLAIN POLYMORPHISM?

5.13) EXPLAIN POLYMORPHISM?

5.14) EXPLAIN POLYMORPHISM?

5.15) EXPLAIN POLYMORPHISM?

5.16) EXPLAIN POLYMORPHISM?

5.17) EXPLAIN POLYMORPHISM?

5.18) EXPLAIN POLYMORPHISM?

5.19) EXPLAIN POLYMORPHISM?

5.20) EXPLAIN POLYMORPHISM?

5.21) EXPLAIN POLYMORPHISM?

5.22) EXPLAIN POLYMORPHISM?

5.23) EXPLAIN POLYMORPHISM?

5.24) EXPLAIN POLYMORPHISM?

5.25) EXPLAIN POLYMORPHISM?

5.26) EXPLAIN POLYMORPHISM?

63)

CAN WE OVERRIDE STATIC METHODS?
No.

64)

EXPLAIN TYPECASTING?

Type Casting :-

One type of data is converted to another type of data is called as type casting.

65)

EXPLAIN PRIMITIVE TYPECASTING?

Primitive Typecasting :- Converting subclasse type ref. variable by using

Converting one primitive data to another primitive data is called as primitive casting.

66)

EXPLAIN AUTO WIDENING & EXPLICIT NARROWING?

Auto Widening :-

Converting one primitive date to another bigger primitive date implicitly is called auto widening.

67)

EXPLAIN AUTO NARROWING

Converting one bigger primitive date to another smaller primitive date by explicitly is called explicit narrowing.

68)

EXPLAIN AUTOUPCASTING & EXPLICIT DOWNGCASTING?

69) Auto upcasting :-

Converting subclasse object to behave like precisely as supclasse type ref. variable by using implicitly is called auto upcasting.

Explicit downcasting :-

Converting supclasse object to behave like subclasse ref. variable explicitly declaring is called explicit downcasting.

70)

CAN WE ACHIEVE W/O OBJECT CASTING W/O INHERITANCE.

No
Because inheritance of class will know its supclasse.

71)

CAN WE ACHIEVE DOWNCASTING W/O UPCASTING?

71) EXPLAIN POLYMORPHISM?

It is a type of oops principle in which a method showing different behaviour in diff. diff. situation.

72) TYPES OF POLYMORPHISM?

2 types →

- Runtime Polymorphism
- CompileTime Polymorphism

73) WHEN DOES JAVA THROWS CLASS CAST EXCEPTION
WHEN DOWNCASTING HAPPENS W/ O DOING
UPCASTING THEN IT THROWS CLASSCAST EXCEPTION.

74) USE OF INSTANCE OF OPERATOR?

IT IS USED TO CHECK WHETHER THE OBJECT IS
OF SPECIFIED TYPE OR NOT.

75) EXPLAIN ABSTRACT METHOD?

A METHOD WHICH HAS ONLY DECLARATION BUT
NO BODY IS CALLED AS ABSTRACT METHOD.
- IT IS DECLARED USING ABSTRACT KEYWORD.

76) EXPLAIN ABSTRACT CLASSES?

A CLASS WHICH IS DECLARED USING ABSTRACT
KEYWORD IS CALLED ABSTRACT CLASS.
IN THIS, BOTH CONCRETE & ABSTRACT METHODS CAN
BE PRESENT.

77) WHY WE CAN'T INSTANTIATE AN ABSTRACT
CLASS?

BECAUSE AN ABSTRACT HAS NOTHING ON ITS OWN.
JVM CAN'T ALLOCATE THE MEMORY OF ABSTRACT
CLASS, SO IT CAN'T CREATE OBJECT OF IT.

78) WHAT IS THE RULE TO BE FOLLOWED BY THE
SUB CLASS OF AN ABSTRACT CLASS?

IT SHOULD EXTENDS TO THE ABSTRACT CLASS
THE METHOD OF SUBCLASS HAS THE SAME SIGNATURE THAT

ABSTRACT METHOD BUT W/ O ABSTRACT KEYWORD.
CAN ABSTRACT CLASS INHERIT FROM ANOTHER
ABSTRACT CLASS?

81) EXPLAIN ABSTRACT CLASSES 100%. ABSTRACT? EXPLAIN?

NO.

BECAUSE IT HAS BOTH ABSTRACT & CONCRETE METHODS.

82) EXPLAIN INTERFACE IN JAVA?

IT IS A KEYWORD DECLARED USING INTERFACE
KEYWORD.

DATA MEMBERS ARE BY DEFAULT STATIC & FINAL.
FUNCTION MEMBERS ARE BY DEFAULT ABSTRACT.

83) DIFFERENCE BETWEEN ABSTRACT CLASS &
INTERFACE?

ABSTRACT CLASS

INTERFACE

IT IS DECLARED USING ABSTRACT KEYWORD.
IT IS DECLARED USING INTERFACE KEYWORD.

THE METHODS OF THIS ARE BY DEFAULT ABSTRACT
CAN BE ABSTRACT OR CONCRETE.

DATA MEMBERS CAN'T HAVE BE BY DEFAULT ABSTRACT
PRIVATE, STATIC & FINAL.
BY DEFAULT FINAL, STATIC
IT HAS CONSTRUCTOR FOR THEM IF IT HASN'T ANY CONSTRUCTORS.

IT IS USED TO ACHIEVE O-OLOGY.
IT IS USED TO ACHIEVE 100% ABSTRACTION.

83) Does ABSTRACT CLASS HAVE CONSTRUCTOR?
Why?
Yes.
Because it is a type of ordinary class which
con. is used to help the ~~class~~ class for
inheriting the properties.

84) Does INTERFACE HAVE CONSTRUCTORS?
No.

85) CAN WE INstantiate AN INTERFACE?
Yes.

86) EXPLAIN ABOUT IMPLEMENTS KEYWORD?
It is used to implement the properties of
interface class to an ordinary class.

87) CAN AN INTERFACE INHERIT FROM ANOTHER
INTERFACE?
Yes.

88) EXPLAIN ABSTRACTION & ITS STEPS & ADVANTAGE
Abstraction :-

It is the process of hiding the implementation
of a function or a method from user.
Step:-
1) Generalisation the common properties of the
implementation class & store it in an interface.
2) Provide the implementation for abstract method
inside the interface.

89) Create the object of implementer class & store
inside interface ref. var.
90) Access the properties of implementer class with
the help of ref. var. of interface.

91) Advantages :-
i) It helps in loose coupling.
ii) We don't need to create class & obj for body of abstract class.
92) CAN WE DEVELOP FINAL METHODS IN INTERFACE?
No.

93) CAN WE DEVELOP FINAL ABSTRACT CLASS?
No.

94) CAN WE DEVELOP STATIC METHOD IN ABSTRACT
CLASS?
No.

95) CAN WE JUST DECLARE A VARIABLE WITHOUT
INITIALIZING?
Yes.

ENCAPSULATION

96) EXPLAIN PACKAGE?
Package :-
It is a collection of java program with
dedicated functionality.

Q4) WHY WE NEED PACKAGES?

We need to differentiate the program so that we can easily locate & use them.

- Code reusability.

Q5) WHAT IS THE DIFFERENT ACCESS LEVELS IN JAVA?

4 levels →
 ↳ Public ↳ Default
 ↳ Protected ↳ Public

Q6) EXPLAIN SINGLETON CLASSES?

It is a class having exactly only 1 object thus that class is called as singletons.

Q7) EXPLAIN ENCAPSULATION?

It is a type of OOPS principle which is used to bind the data & functions into a single unit.

Q8) EXPLAIN THE JAVA BEAN CLASSES?

Data members are private.
It has a parameterized constructor.
It is declared public getter & setter methods.

Adv
 ↳ data access opn
 ↳ state transfer opn

LIBRARIES

Q9) WHICH IS THE SUPERMOST CLASS OF ALL THE OBJECTS?

Object class

Q10) EXPLAIN `toString()`, `hashCode()`, `equals()`.

↳ It is used to return unique string.
↳ It is used for generally printing purpose.

`hashCode()` ↳

↳ It is used to return unique integer value which is assigned by JVM to an object.

`equals()` ↳

↳ It is used to return boolean value to compare the contents of 2 objects.

Q11) CAN SUBCLASS OVERRIDE `toString()`, `hashCode()`, `equals()`?

↳

Q12) EXPLAIN FINAL METHOD IN JAVA?

↳ To make method to be constant we go for final method.

↳ We can't override a final method.

Q13) EXPLAIN FINAL CLASSES?

↳ We can't inherit a final class.

104) USE OF FINALIZED ?

It is used to do some cleanup oprn in the object before its destroy.

105) USE OF CLONE()

When we want to create the exact duplicate of existing class we go for clone.

106) WHAT IS A MARKER INTERFACE ?

An empty interface is called as marker interface to give the permission to java obj. to change its code to byte.

107) EXPLAIN ALL MARKER INTERFACE IN JAVA

- 1) Clonable - To create exactly dup. of a class (javaway)
- 2) Serializable - To give the permission for saving the state of object (javaway)
- 3) Random Access (java.rmi)

↳ To give facility for access of unclss

108) EXPLAIN ABOUT ARRAY ?

It is collection of similar types of data under single reference.

109) EXPLAIN PRIMITIVE ARRAYS & NON-PRIMITIVE

CLASS TYPE OR DERIVED) ARRAY ?

Primitive Array :

It is used to store only primitive data.

Derived array :

It is used to store only non-primitive data.

110) HOW STRINGS ARE IMMUTABLE ?

Strings are immutable because its datamember can't be changed further.

111) CAN WE INHERIT THE STRING CLASS ?

No

112) EXPLAIN THE BEHAVIOR OF TOSTRING(), HASHCODE(), EQUALS(),

toString() :-

It is used to return the unique integer value

which is given by JVM to specific class.

equals() :-

It is used to compare b/w two strings based on content of object.

↳ return boolean value

113) EXPLAIN STRING POOL AREA ?

It is the area of memory where class loader loads the static members of class

114) DIFFERENCE BETWEEN '==' & .EQUALS(),

==

↳ It is used to compare ↳ It is used to compare address of object. the contents of object.

三

DIFFERENCE STRING, STRING, BULDEK & STRINGS

String	StringBuilder	StringBuffer
immutable	is mutable	is mutable
Thread safe	is Not Thread safe	is Not Thread safe
Override only String ()	is Override only	is Override only
is Reusable	is Reusable	is Reusable
to string ()	to string ()	to string ().
is Asynchronous	is Not synchronized	is synchronized
is Reentrant ()	is Reentrant ()	is Reentrant ()

16) EXPLAIN ABOUT COLLECTION

117) EXPLAIN LST, SET & QUEUE ?

list Set Queue

i) index based col \Rightarrow not index based \Rightarrow not index based

ii) insertion order preserved \Rightarrow Insertion order \Rightarrow Insertion order

iii) dup. & null are \Rightarrow dup. are not \Rightarrow dup value allowed \Rightarrow allowed

value allowed

三

EXPLAIN THE FEATURES OF ARRAYS, LIST, LINKEDLIST, PRIORITY QUEUE WITH A PROGRAM?

Ex.

```
int id;
String name;
double marks;
```

This name, name;
this works, works;
a greenide.

public string ToString() {

return `id` + " " + `name` + " " + `workid` ?

PSVY (String [] array) 9

ArrayList \rightarrow new Area

١٢. آنکه میگویند این اینست که

S 33, New 313.

5. 842 new s (14, "EP", 2.9);

al. add (S1);

ad. 800 (120)

al. *Gold Csy*

for Object on

11.00 50P (50b)

Enkele stijlen:

155

```
for (Object ob : l1) {
    ? sop (ob);
}
PriorityQueue <P> = new PriorityQueue ();
P. add (l1);
P. size (); 4
for (int i = 0; i < P.size (); i++) {
    sop (P.peek ());
}
3 P.size (); 0
```

```
120) EXPLAIN HASHSET, LINKEDSET & TREESET  
WITH A PROGRAM?  
class S {
    int id;
    double marks;
}
public S <int id, double marks>?
    this.id, id;
    this.marks, marks;
}
@ overide
```

```
public String toString () {
    return id + " " + marks;
}
public class M {
    String C (String l1, String l2) {
        M m = new M ();
        m.s1 = l1;
        m.s2 = l2;
        m.s3 = l1 + " " + l2;
        m.s4 = l2 + " " + l1;
        return m;
    }
}
```

```
120) NEED & DIFFERENCE BETWEEN COMPARABLE  
& COMPARATOR INTERFACES  
Comparatore
```

```
javLang package. & java.util package  
so that this object is get has the capability of  
capable of comparing 2 diff. object  
itself with another  
object.
```

EXPLAIN GENERAL & HOW TO APPLY IT TO

COLLECTION ?
In order to make the coll' compile time
type safe we prefer collection, i.e only 1
type of obj. want to store in 1 coll.

Collection <sp:type> ref = new Collection();
var

Explain Maps?
Maps stores the element in the form of key value pair.

Mr. Pitt (2, Valence) 81

```
hm.put (54, value5); value1  
sop (hm.get(1)); value2  
sop (hm.get(2)); value3
```

SOPC km. get (3a); null
SOPC km. get (3a); values

126) EXPLAIN BOXING & UNBOXING?

The process of converting a primitive value to a java project

The process of concatenating a java project back to primitive value.

123) WHAT IS WRAPPER CLASS ? USES ?

WHAT IS WRAPPER CLASS ? USES ?
Java supports 8 primitive datatype but it is not 100% type language.
To overcome this we make the use of wrapper class.

Ways :-
It helps to represent or convert primitive value to java object & vice versa.

EXPLAIN EXCEPTIONS IN JAVA ?
It is an event triggered by the risky lines of code present in the program.

Ques. HOW TO HANDLE EXCEPTION ?

Ans. $\frac{1}{2}$ try-catch
 $\frac{2}{2}$ Using throws keyword

Q131) DIFFERENCE & EXPLAIN CHECKED & UNCHECKED EXCEPTION ?

ANSWER:

1) CHECKED UNCHECKED

↳ subclass of exception ↳ Runtime exception &
other than run time it subclass

exception.
i) known @ compilation ii) known & occurs @
 occurred @ runtime runtime
iii) can't propagate iv) can propagate
 implicitly implicitly.

13.2) DIFFERENCE BETWEEN THROW & THROWS?

Throw

↳ Used to generate an exception and to propagate the exception of obj by the programmer.

method.

↳ Written in method body.

↳ Written in method decl.

↳ declare n'th. of exception

↳ followed by obj. of class.

name.

13.3) EXPLAIN GENERIC HANDLER & SPECIFIC HANDLER

Generic

↳ It is type of exception. It is a type of handler which is used to handle all type of specific type of exception.

↳ It is type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

↳ It is used to handle all type of exception.

13.4) DIFFERENCE BETWEEN FINALLY, FINALLY, FINALLY

Final

Finally

Finally

To make a resource to make some code to be up-keeping an object as constant we use final keyword.

To perform some clean up for final object executed before destroying keyword.

In any scenario Garbage collector uses for finally invoke finally block.

13.5) EXPLAIN HOW TO WRITE THE DATA INTO A

TEXT FILE?

Class M is Psvm

String data = "Java";

FileWriter fw = null;

try {

fw = new FileWriter("path");

fw.write(data);

fw.flush();

fw.close();

catch (Exception e) {

e.printStackTrace();

13.6) USE OF SCANNER CLASS

It is used to read data from the

user from keyboard.

139

HOW TO READ THE DATA FROM CONSOLE?

140

Main thread is created by JVM for execution purpose.

```
class Main {
    public static void main(String[] args) {
        FileReader fr = null;
        try {
            fr = new FileReader("path");
            int a = fr.read();
            System.out.println(a);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

↳ We can create undefined thread with the help of creating thread class

↳ using runnable interface

```
class MyThread implements Runnable {
    public void run() {
        System.out.println("Hello");
    }
}
```

↳ DIFFERENT WAYS OF CREATING THREADS?

1) Extending thread class

2) Implementing runnable interface.

140 HOW TO READ THE STRING FROM CONSOLE?

Using scanner class

140 EXPLAIN SERIALIZATION & DESERIALIZATION

Serilization :-

The process of converting primitive data to java project to ~~String~~ of bytes.

140 EXPLAIN start() & Run() ?

start() :-

It is used to start the execution by implicitly calling run() to achieve multithreading.

run() :-

It is used to execute a singutask.

Deserialization :-

The process of converting stream of bytes back to java project.

140 EXPLAIN THREAD & ITS PROPERTIES?

It is an execution instance which is used to achieve multitasking.

Main thread is the thread which is created by memory.

↳ Declaring all methods of class with synchronized

140 EXPLAIN A THREAD SAFE CLASS & HOW TO CREATE IT?

Thread safe class is used to avoid the thread collision. To make it:-

↳ Make the class immutable

↳ Declaring all methods of class with synchronized

148) EXPLAIN THREAD DEADLOCK & HOW TO AVOID IT?

It is a situation where the thread got the access of thread safe obj but can't execute the task & will not release the key to others. Go to this other thread are waiting for key to amount time.

149) EXPLAIN INTERTHREAD COMMUNICATION & ITS METHODS?

To break the thread deadlock we go for Externthread communication.

5 methods → wait();
wait(long ms);
wait(int a, long ms);
notify();
notifyAll();

150) DIFFERENCE BETWEEN ABSTRACT CLASS & INTERFACE?
Refer 82

151) DIFFERENCE BETWEEN THIS() & SUPER()?
Refer 57

152) DIFFERENCE BETWEEN THIS & SUPER KEYWORD.
Refer 58

153) EXPLAIN INHERITANCE & ITS TYPES? GIVE A REAL TIME EXAMPLE
Refer 52

Ex. class ~~Calculator~~ SAP ("calculating");
public class ~~Calculator~~ Mobile {
public void ~~display~~("Showing date & time");
class mobile {

public void ~~display~~ ("Showing wallpaper");
class H {
public void ~~display~~ (String [] args) {
calculator c1 = new calculator();
c1. c1();
c1. m1();
calculator c2 = new calculator();
c2. c2();
c2. m2();
}

154) DIFFERENCE BETWEEN METHOD OVERLOADING & OVERRIDING?
Method Overloading
↳ Multiple methods having same name but diff. some redeclaration but with their args in diff. with their method overloading. implementation is called overriding.

↳ It can be static/non-static
i) It can't be static
final.
ii) Inheritance not necessary
iii) Inheritance is mandatory
iv) It is a form of compilation polymorphism
v) It is a form of runtime polymorphism

15) WHY MIGRATION IS NOT POSSIBLE?

159) DIFFERENCE BETWEEN LIST, SET, QUEUE?

↳ In multiple inheritance, super statement

of subclass get confused which superclass
con. to inherit call & which class to inherit.
By this subclass will get confused
to select the object class behaviour. It is
known as 'multiple inheritance'.

15) Explain upcasting & downcasting?

Q6) Explain Polymorphism & its types & example
(QF Real Time)

157) EXPLAIN ABSTRACTION & HOW TO ACHIEVE IT ?

Refer Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36 Q37 Q38 Q39 Q40 Q41 Q42 Q43 Q44 Q45 Q46 Q47 Q48 Q49 Q50 Q51 Q52 Q53 Q54 Q55 Q56 Q57 Q58 Q59 Q60 Q61 Q62 Q63 Q64 Q65 Q66 Q67 Q68 Q69 Q70 Q71 Q72 Q73 Q74 Q75 Q76 Q77 Q78 Q79 Q80 Q81 Q82 Q83 Q84 Q85 Q86 Q87 Q88 Q89 Q90 Q91 Q92 Q93 Q94 Q95 Q96 Q97 Q98 Q99 Q100 Q101 Q102 Q103 Q104 Q105 Q106 Q107 Q108 Q109 Q110 Q111 Q112 Q113 Q114 Q115 Q116 Q117 Q118 Q119 Q120 Q121 Q122 Q123 Q124 Q125 Q126 Q127 Q128 Q129 Q130 Q131 Q132 Q133 Q134 Q135 Q136 Q137 Q138 Q139 Q140 Q141 Q142 Q143 Q144 Q145 Q146 Q147 Q148 Q149 Q150 Q151 Q152 Q153 Q154 Q155 Q156 Q157 Q158 Q159 Q160 Q161 Q162 Q163 Q164 Q165 Q166 Q167 Q168 Q169 Q170 Q171 Q172 Q173 Q174 Q175 Q176 Q177 Q178 Q179 Q180 Q181 Q182 Q183 Q184 Q185 Q186 Q187 Q188 Q189 Q190 Q191 Q192 Q193 Q194 Q195 Q196 Q197 Q198 Q199 Q200 Q201 Q202 Q203 Q204 Q205 Q206 Q207 Q208 Q209 Q210 Q211 Q212 Q213 Q214 Q215 Q216 Q217 Q218 Q219 Q220 Q221 Q222 Q223 Q224 Q225 Q226 Q227 Q228 Q229 Q230 Q231 Q232 Q233 Q234 Q235 Q236 Q237 Q238 Q239 Q240 Q241 Q242 Q243 Q244 Q245 Q246 Q247 Q248 Q249 Q250 Q251 Q252 Q253 Q254 Q255 Q256 Q257 Q258 Q259 Q260 Q261 Q262 Q263 Q264 Q265 Q266 Q267 Q268 Q269 Q270 Q271 Q272 Q273 Q274 Q275 Q276 Q277 Q278 Q279 Q280 Q281 Q282 Q283 Q284 Q285 Q286 Q287 Q288 Q289 Q290 Q291 Q292 Q293 Q294 Q295 Q296 Q297 Q298 Q299 Q300 Q301 Q302 Q303 Q304 Q305 Q306 Q307 Q308 Q309 Q310 Q311 Q312 Q313 Q314 Q315 Q316 Q317 Q318 Q319 Q320 Q321 Q322 Q323 Q324 Q325 Q326 Q327 Q328 Q329 Q330 Q331 Q332 Q333 Q334 Q335 Q336 Q337 Q338 Q339 Q340 Q341 Q342 Q343 Q344 Q345 Q346 Q347 Q348 Q349 Q350 Q351 Q352 Q353 Q354 Q355 Q356 Q357 Q358 Q359 Q360 Q361 Q362 Q363 Q364 Q365 Q366 Q367 Q368 Q369 Q370 Q371 Q372 Q373 Q374 Q375 Q376 Q377 Q378 Q379 Q380 Q381 Q382 Q383 Q384 Q385 Q386 Q387 Q388 Q389 Q390 Q391 Q392 Q393 Q394 Q395 Q396 Q397 Q398 Q399 Q400 Q401 Q402 Q403 Q404 Q405 Q406 Q407 Q408 Q409 Q410 Q411 Q412 Q413 Q414 Q415 Q416 Q417 Q418 Q419 Q420 Q421 Q422 Q423 Q424 Q425 Q426 Q427 Q428 Q429 Q430 Q431 Q432 Q433 Q434 Q435 Q436 Q437 Q438 Q439 Q440 Q441 Q442 Q443 Q444 Q445 Q446 Q447 Q448 Q449 Q450 Q451 Q452 Q453 Q454 Q455 Q456 Q457 Q458 Q459 Q460 Q461 Q462 Q463 Q464 Q465 Q466 Q467 Q468 Q469 Q470 Q471 Q472 Q473 Q474 Q475 Q476 Q477 Q478 Q479 Q480 Q481 Q482 Q483 Q484 Q485 Q486 Q487 Q488 Q489 Q490 Q491 Q492 Q493 Q494 Q495 Q496 Q497 Q498 Q499 Q500 Q501 Q502 Q503 Q504 Q505 Q506 Q507 Q508 Q509 Q510 Q511 Q512 Q513 Q514 Q515 Q516 Q517 Q518 Q519 Q520 Q521 Q522 Q523 Q524 Q525 Q526 Q527 Q528 Q529 Q530 Q531 Q532 Q533 Q534 Q535 Q536 Q537 Q538 Q539 Q540 Q541 Q542 Q543 Q544 Q545 Q546 Q547 Q548 Q549 Q550 Q551 Q552 Q553 Q554 Q555 Q556 Q557 Q558 Q559 Q560 Q561 Q562 Q563 Q564 Q565 Q566 Q567 Q568 Q569 Q570 Q571 Q572 Q573 Q574 Q575 Q576 Q577 Q578 Q579 Q580 Q581 Q582 Q583 Q584 Q585 Q586 Q587 Q588 Q589 Q590 Q591 Q592 Q593 Q594 Q595 Q596 Q597 Q598 Q599 Q600 Q601 Q602 Q603 Q604 Q605 Q606 Q607 Q608 Q609 Q610 Q611 Q612 Q613 Q614 Q615 Q616 Q617 Q618 Q619 Q620 Q621 Q622 Q623 Q624 Q625 Q626 Q627 Q628 Q629 Q630 Q631 Q632 Q633 Q634 Q635 Q636 Q637 Q638 Q639 Q640 Q641 Q642 Q643 Q644 Q645 Q646 Q647 Q648 Q649 Q650 Q651 Q652 Q653 Q654 Q655 Q656 Q657 Q658 Q659 Q660 Q661 Q662 Q663 Q664 Q665 Q666 Q667 Q668 Q669 Q670 Q671 Q672 Q673 Q674 Q675 Q676 Q677 Q678 Q679 Q680 Q681 Q682 Q683 Q684 Q685 Q686 Q687 Q688 Q689 Q690 Q691 Q692 Q693 Q694 Q695 Q696 Q697 Q698 Q699 Q700 Q701 Q702 Q703 Q704 Q705 Q706 Q707 Q708 Q709 Q710 Q711 Q712 Q713 Q714 Q715 Q716 Q717 Q718 Q719 Q720 Q721 Q722 Q723 Q724 Q725 Q726 Q727 Q728 Q729 Q730 Q731 Q732 Q733 Q734 Q735 Q736 Q737 Q738 Q739 Q740 Q741 Q742 Q743 Q744 Q745 Q746 Q747 Q748 Q749 Q750 Q751 Q752 Q753 Q754 Q755 Q756 Q757 Q758 Q759 Q760 Q761 Q762 Q763 Q764 Q765 Q766 Q767 Q768 Q769 Q770 Q771 Q772 Q773 Q774 Q775 Q776 Q777 Q778 Q779 Q780 Q781 Q782 Q783 Q784 Q785 Q786 Q787 Q788 Q789 Q790 Q791 Q792 Q793 Q794 Q795 Q796 Q797 Q798 Q799 Q800 Q801 Q802 Q803 Q804 Q805 Q806 Q807 Q808 Q809 Q810 Q811 Q812 Q813 Q814 Q815 Q816 Q817 Q818 Q819 Q820 Q821 Q822 Q823 Q824 Q825 Q826 Q827 Q828 Q829 Q830 Q831 Q832 Q833 Q834 Q835 Q836 Q837 Q838 Q839 Q840 Q841 Q842 Q843 Q844 Q845 Q846 Q847 Q848 Q849 Q850 Q851 Q852 Q853 Q854 Q855 Q856 Q857 Q858 Q859 Q860 Q861 Q862 Q863 Q864 Q865 Q866 Q867 Q868 Q869 Q870 Q871 Q872 Q873 Q874 Q875 Q876 Q877 Q878 Q879 Q880 Q881 Q882 Q883 Q884 Q885 Q886 Q887 Q888 Q889 Q890 Q891 Q892 Q893 Q894 Q895 Q896 Q897 Q898 Q899 Q900 Q901 Q902 Q903 Q904 Q905 Q906 Q907 Q908 Q909 Q910 Q911 Q912 Q913 Q914 Q915 Q916 Q917 Q918 Q919 Q920 Q921 Q922 Q923 Q924 Q925 Q926 Q927 Q928 Q929 Q930 Q931 Q932 Q933 Q934 Q935 Q936 Q937 Q938 Q939 Q940 Q941 Q942 Q943 Q944 Q945 Q946 Q947 Q948 Q949 Q950 Q951 Q952 Q953 Q954 Q955 Q956 Q957 Q958 Q959 Q960 Q961 Q962 Q963 Q964 Q965 Q966 Q967 Q968 Q969 Q970 Q971 Q972 Q973 Q974 Q975 Q976 Q977 Q978 Q979 Q980 Q981 Q982 Q983 Q984 Q985 Q986 Q987 Q988 Q989 Q990 Q991 Q992 Q993 Q994 Q995 Q996 Q997 Q998 Q999 Q1000 Q1001 Q1002 Q1003 Q1004 Q1005 Q1006 Q1007 Q1008 Q1009 Q10010 Q10011 Q10012 Q10013 Q10014 Q10015 Q10016 Q10017 Q10018 Q10019 Q10020 Q10021 Q10022 Q10023 Q10024 Q10025 Q10026 Q10027 Q10028 Q10029 Q10030 Q10031 Q10032 Q10033 Q10034 Q10035 Q10036 Q10037 Q10038 Q10039 Q10040 Q10041 Q10042 Q10043 Q10044 Q10045 Q10046 Q10047 Q10048 Q10049 Q10050 Q10051 Q10052 Q10053 Q10054 Q10055 Q10056 Q10057 Q10058 Q10059 Q10060 Q10061 Q10062 Q10063 Q10064 Q10065 Q10066 Q10067 Q10068 Q10069 Q10070 Q10071 Q10072 Q10073 Q10074 Q10075 Q10076 Q10077 Q10078 Q10079 Q10080 Q10081 Q10082 Q10083 Q10084 Q10085 Q10086 Q10087 Q10088 Q10089 Q10090 Q10091 Q10092 Q10093 Q10094 Q10095 Q10096 Q10097 Q10098 Q10099 Q100100 Q100101 Q100102 Q100103 Q100104 Q100105 Q100106 Q100107 Q100108 Q100109 Q100110 Q100111 Q100112 Q100113 Q100114 Q100115 Q100116 Q100117 Q100118 Q100119 Q100120 Q100121 Q100122 Q100123 Q100124 Q100125 Q100126 Q100127 Q100128 Q100129 Q100130 Q100131 Q100132 Q100133 Q100134 Q100135 Q100136 Q100137 Q100138 Q100139 Q100140 Q100141 Q100142 Q100143 Q100144 Q100145 Q100146 Q100147 Q100148 Q100149 Q100150 Q100151 Q100152 Q100153 Q100154 Q100155 Q100156 Q100157 Q100158 Q100159 Q100160 Q100161 Q100162 Q100163 Q100164 Q100165 Q100166 Q100167 Q100168 Q100169 Q100170 Q100171 Q100172 Q100173 Q100174 Q100175 Q100176 Q100177 Q100178 Q100179 Q100180 Q100181 Q100182 Q100183 Q100184 Q100185 Q100186 Q100187 Q100188 Q100189 Q100190 Q100191 Q100192 Q100193 Q100194 Q100195 Q100196 Q100197 Q100198 Q100199 Q100200 Q100201 Q1

16118509 LIST OUT THE DIFFERENCE BETWEEN

Refer 136 final, finely, finalized

Explain Garbage Collector ?

to ~~can~~ my differences b/w. after execution of program, I my calls Garbage collector.

—T, QMDE?

Rufen (118)

Abstraction ?
Encapsulation

→ grouping of members of a class of objects
the data members & implementation of a function into a single unit. a function or class form.

iii) It is accessed using interface \rightarrow It is accessed using abstract class. \rightarrow access specifier
iv) It can't be private, \rightarrow It can be private, static & final.

165) What is the reverse & string?

```

class Solution {
public:
    string reverseString(string s) {
        int i = 0, j = s.size() - 1;
        while (i < j) {
            swap(s[i], s[j]);
            i++;
            j--;
        }
        return s;
    }
};

```

```
for (int i = 0; i <avelengths - 1; i++)  
    copy(&res[i], &res[i + 1]);
```

164) WAP TO HANDLE THE EXCEPTION USING CATCH BLOCK
class MyException{
String Target;
try {
System.out.println("try");
} catch(ArithmaticException e){
System.out.println("catch");
e.printStackTrace();
}
}

165)

WAP TO COUNT NUMBERS IN A GIVEN STRING,

"hey12345they"?

```
class N { public static void main (String [] args) {
```

```
    String s1 = "hey123457hey";
```

```
    int count = 0;
```

```
    for (int i = 0; i < s1.length() - 1; i++) {
```

```
        if (s1[i] >= '0' && s1[i] <= '9')
```

```
            count++;
```

```
}
```

```
    System.out.println(count);
```

166)

WAP TO REVERSE THE WORDS OF GIVEN STRING?

```
class N { public static void main (String [] args) {
```

```
    String s = "name is james";
```

```
    String rev = " ";
```

```
    String [] str = s.split(" ");
```

```
    for (int i = 0; i < s.length() - 1; i++)
```

```
{
```

```
        String s1 = str[i]; String s2 = str[s.length() - 1 - i];
```

```
        if (i == 0) {
```

```
            replace;
```

```
            String rev = rev + replace(s1, s2);
```

```
}
```

```
        if (i == s.length() - 1)
```

```
            rev = rev + replace(s2, s1);
```

```
{
```

```
    System.out.println(rev);
```

```
{
```

```
}
```