

JAVA

Introduction

1)What is Language?

→ Communication media through which we can develop communicate with each other.

2)What is Programming Language?

→ A programming language is an intermediate between user and computer system.

3)Which are the characteristics of Programming language?

→ i) **Syntax**(Rules given by programming language)

ii) **Semantics**(Structure or Meaning of the code)

4)What is Java Language?

→ Java is a high-level object oriented secure, robust and platform independent programming language.

- By using Java language, we can develop various applications, which are as follows.

i) **Software or Standalone application**

ii) **Website or Webapplication**

iii) **Mobile application**

iv) **Enterprise application**

v) **Distributed application and Gaming application**

5)Difference between Software or Website.

→ Whenever client data is private we should suggest, software or standalone application is applied.

- In the Software Application, maintenance is a challenge, so we can upload the private data program at server location with valid username and password, in client can view, update the record, in a personal profile with the help of Web application.

6)Which are the flavors of Java language?

→ Java language provides 4 types of flavors.

i) **JSE (Java Standard Edition)** - It is also known as Core Java, By using JSE we can develop Standalone application or Software.

ii) **JEE (Java Enterprise Edition)** - It is also known as Advanced Java, By using JEE we can develop Websites.

iii)JME(Java Micro Edition)-It is also known as Android Java,by using JME we can develop mobile application.

iv)JavaFx-It is used to develop GUI application,It is outdated you have to develope JUI application now we are using NodeJS or react.

7)History of Java.

→-Java was developed by Sun-Microsystem but on 27th January 2010,Java was overtaken by Oracle Corporation so now,Java is the product of oracle corporation.

-Java was developed by James Gosling.

8)What is function?

→-A function is a self-defined block which is used for performing some operation calculate or printing the Data.

9)Types of function.

→-function can be divided into two types.

i)**Predefined or Built in function**(The function which is written by language creator or user for some specific purpose is called predefined functon.

ii)**User-Defined function**(The function which are defining by user for some specific task are called user-defined function.

10)Advantages of function.

→i)**Modularity**(Dividing the bigger task into number of smaller task)

ii)**Easy to understand**(Once the task is divided into number of independent modules then it is easy to understand the entire module.

iii)**Reusability**(We can reuse a particular module for 'n' number of times.

iv)**Easy debugging**(Each module is isolated from another module so the debugging is easy because we can debug only one module where we have syntax or semantics error.

11)Why we pass parameter/argument to a function?

→We should pass parameter to a function for getting more information regarding the function.

12)Why function are called Method in java?

→In c++ language,there is a facility to write a function inside the class as well as outside of the class by using

Scope resolution operator(::) but in java we can write a function inside the class only,we can't define a Function outside of the class,that is the reason function are called Method in java.

13)What is platform independency in java?

→ C and C++ programs are platform dependent program that means the .exe file created on one machine will not be executed on the another machine if the system configuration is different.

14)What is the role of java compiler?

→ i) Syntax Verification
ii) Verify the Compatibility issues(L.H.S=R.H.S)
iii) Will convert source code into byte code.

15)Why Java is a platform independent language?

→ Whenever we write a java program, the extension of java program must be .java.
Now this .java file we submit to java compiler(javac) for compilation process. After successful compilation the compiler will generate a very special machine code file i.e. Class file(also known as byte code). Now this .class file we submit to JVM for execution purpose.
The role of JVM is to load and execute the .class file. Here JVM plays a major role because it converts the .class file into appropriate machine code instruction(operating system format) so java becomes platform Independent language and it is highly suitable for website development.
We have different JVM for different operating system that means JVM is platform dependent technology. Where as java is platform independent technology.
JVM internally contains an interpreter so it executes the code line by line. It is written in 'C' language Hence platform dependent.

16)What is the difference between bit code and byte code?

→ Bit code is directly understood by operating system but on the other side byte code is understood by JVM, JVM is going to convert this byte code into appropriate machine understandable format.

17)What is JVM?

→ It stands for Java Virtual Machine, it is responsible to load and execute the .class file. It internally contains an interpreter and it is written by using C language.
Java is platform dependent, each O.S contains its specific JVM, JVM plays a major role because it converts .class file into appropriate operating system format.

18)Difference between Compiler and Interpreter.



Compiler	Interpreter
i)It scans the entire program once and converts source code into byte code.	i)It scans the program line by line for generating the output.
ii)It displays all the errors and warnings at a time so debugging is slow.	ii)It displays the exception only one line at a time so debugging is fast.
iii)It converts source code into byte code so separate memory is required to hold the byte code.	iii)JVM will not generate any intermediate file it takes one line byte code and convert it into output so, separate memory is not required.
iv)Once we resolve all the errors and warnings then execution is fast.	iv)Interpreter executes line by line so, if we make a mistake at line number 6 then after resolving the issues, it will again start the execution from line number 1 so it's slow in nature.
v)The languages like C,C++,Java,C# and so on are using compiler.	v)The languages like Python,Javascript,HTML, Java and so on are using Interpreter.

19)What is JDK?

- It stands for Java Development Kit, it is a developer version that means by using JDK we can develop as well as execute our java programs.
- In order to develop and execute it supports various JDK tools which are as follows.
- i)**javac**(Java compiler,responsible for compilation)
 - ii)**java**(Java launcher,responsible for executing java program)
 - iii)**jdb**(Java debugger,for debugging purpose)
 - iv)**javap**(Java profiler,To get the details of the class)
 - v)**jconsole**(Java console,to display the output in the console)
 - vi)**Javadoc**(Java documentation,for generating java documentation)

20)What is JRE?

- It stands for Java Runtime Environment, It is a client version so by using JRE we can only execute our java programs.

21)What is JIT Compiler?

→-As we know compiler is slow in nature so to boost up the java execution,we have JIT(Just In Time) compiler Supports.

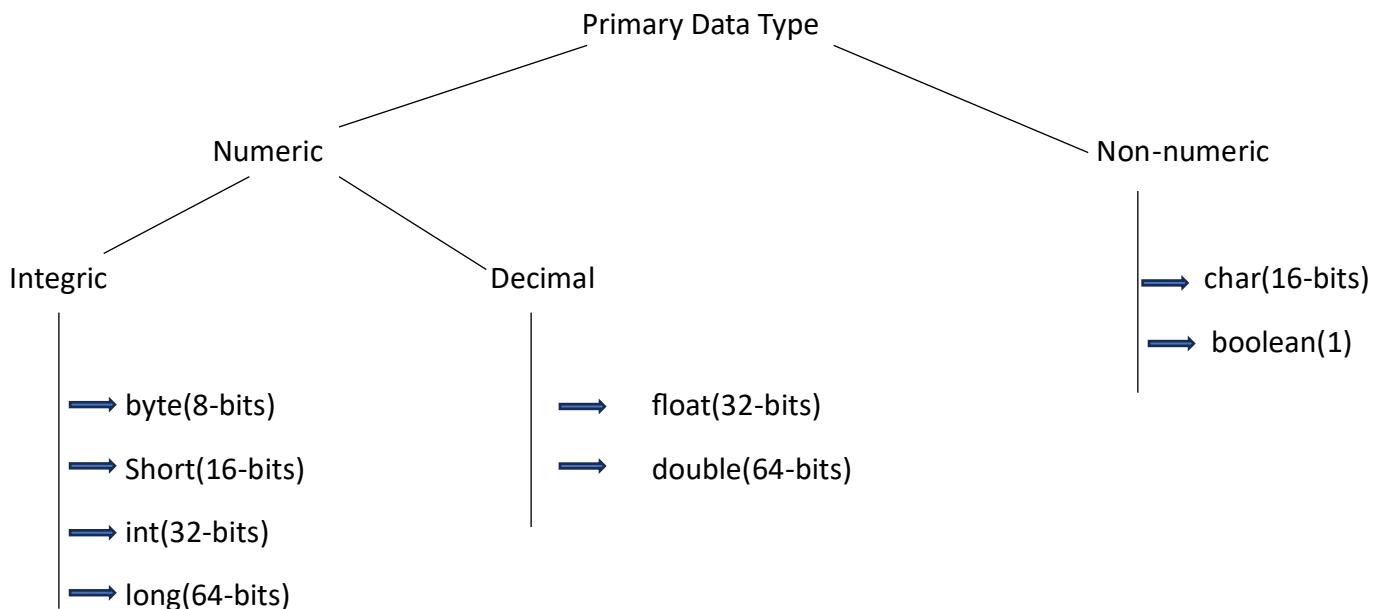
-It holds the repeated code instruction and native code instruction,it will directly provide these two Instruction at a time of line by line execution so our interpreter executes the code in more efficient way hence the overall execution become very fast.

22)What is datatype in java?

→-A datatype describes,what type of value the variable will hold.

-Two types of datatypes.

i)Primitive data types



ii)Reference data types(Class,array,interface,record,enum and so on)

23)What is the difference between Statically typed and Dynamically typed language?

→-The languages where data type is compulsory before initialization of a variable are called statically typed language.

-The languages where data type is not compulsory and it is optional before initialization of a variable then it is called dynamically typed language.

24)What is Comments in java?

→Comments are used to enhance the readability of the code.it is ignored by the compiler.

i)Single Line Comment

ii)Multi line Comment

iii)Documentation Comment.

25)Description of main() method.

→-**public**(public is an access modifier in java.The main method must be declared as public otherwise JVM cannot execute our main method or in other words JVM can't enter inside the main method for execution of the program.If main method is not declared as public then program will compile but it will not be execute in JVM)

-**Static**(Because the method belongs to the class,so object creation is required so that's reason main Method must be static.)

-**Void**(It is a keyword.it means no return type,whenever we define any method in java and if we don't want to return any kind of value from that particular method then we should write void before the name of the method.)

-**main()**-It is a user defined method because a user is responsible to define some logic inside the main method,as well as the execution of the program ends with main() method only.

26)Can we write multiple method with same name?

→Yes,we can write multiple method with same name but parameter must be different otherwise code will not compile.

27)Why the main method of java accepts String array as a parameter?

→String is a collection of alpha-numeric character so it can accept all different kind of values.Java software people has provided String array as a parameter so it can accept multiple values of different type,that means providing more wider scope to accept heterogeneous types of values .

28)What is Command Line Argument.

→Whenever we pass any argument to the main method then it is called Command Line Argument.

29)How to convert String into integer value?

→There is a predefined class called Integer available in java.lang.package,It provides a predefined static Method called parseInt(String x) which accept a single string type parameter and convert this string into int type because the return type of this parseInt(String x) method is int type.

30)What is Tokens in java?

→-Tokens is the smallest unit of the program which is identified by the compiler.
-Tokens is divided into 5 types in java.

i)Keyword

ii)Identifier

iii)Literal

iv)Punctuators(Seperators)

v)Operator.

31)What is Literals in java?

→-Any constant which are assigning to variable is called Literals

-In java we have 5 types of Literals.

i)Integral Literal

ii)floating point Literal

iii)Boolean Literal

iv)Character Literal

v)String Literal

32)Is java is a pure Object Oriented Langauge?

→-No,Java is not a pure object-oriented language because it is accepting primary data types,Actually any language which accepts primary data type is not a pure object-oriented language.

-Only object are moving in the network but not the primary data type so java has introduced Wrapper class Concept to convert the primary data types into corresponding wrapper object.

33)Var keyword in java.

→Var keyword is introduced from java 10v . It can be used inside the method only . it can be initialized in the same line where we are declaring the variable with var keyword,it is also known as local variable type inference.

34)Local Variable in java?

→If we declare a variable inside a method or block or constructor then it is called local variable.

35)Why we can't use a local variable outside of the method or block or constructor?

→-In java,Every methods are executed in a special memory called Stack memory.

-Stacks memory works on LIFO(Last In First Out) basis . In java,whenever we call a method then a separate stack will be created for each and every method.

-Once the method execution is over then the corresponding method statck frame will also be deleted from stack area,that is the reason we can't use local variable outside of the method.

36)Member access operator(.)

→ It is called member access operator, by using this we can access the member of the class.

37)New keyword.

→ -It is also an operator.

-It is used to create the object and initialize the non-static member with default value.

38)What is Reference variable?

→ In java, if a variable is declared with class name then it is called reference variable.

Object Oriented Programming(OOPs)

39)What is an Object?

→ An object is a physical entity, which exists in the real world.

40)Advantages of OOPs.

→ i) **Modularity** (Dividing the bigger task into smaller task)

ii) **Reusability** (We can reuse the component so many times)

iii) **Flexibility** (Easy to maintain)

41)Features of OOPs.

→ i) **Class**

ii) **Object**

iii) **Abstraction**

iv) **Encapsulation**

v) **Inheritance**

vi) **Polymorphism**

42)What is Class?

→ -A class is a model/blueprint/template/prototype for creating the object.

-A class is a logical entity , which does not take any memory.

-A class is a user – defined data type which contains data member member function.

43)What is instance or Non-static variable?

→ It is a class level variable so it has default value, If a non-static variable is defined inside a class but outside of the method then it is called instance variable.

44)Parameter Variable.

→ It is a method level variable hence does not have default value.

-If a variable is declared inside a method parameter then it is called Parameter Variable.

45)What is Constructor in java?

→ If the name of the class and name of the method both are exactly same and it does not contain any return Then it is called Constructor.

46)Types of Constructor in java.

→ i) Default no argument constructor.

ii) Non-parameterized constructor.

iii) Parameterized constructor.

47)What is setter and getter in java?

→ i) Setter (To modify the existing object data [only one data at a time]).

ii) Getter (To read/retrieve private data value outside of BLC class [Reading Operation]).

48)What is factory method?

→ If a method return type is class name means it is returning the object of the class then it is called factory method.

49)What is Shallow Copy?

→ In Shallow copy , only one object is created but the same object will be referred by multiple reference Variable.

-If we modify the object properties by any of the reference variable then original object is also modified.

50)What is Deep Copy?

→ In Deep copy two different object will be created, the 2nd object will copy the content of first object.

-If we modify the object by using reference variable then only one object will be modified.

51)What is Pass by Value in java?

→ Pass by value means we are sending the copy of original data to the method.

52)What is Garbage Collector in java?

→ Garbage Collector is a program in Java that automatically finds and removes objects from memory that are no longer used by the program.

53)How many ways we can make an object eligible for Garbage Collector?

- i) Assigning null literal to existing reference variable.
- ii) Creating an object inside a method.
- iii) Assigning new object to the old existing reference variable.

54)What is Constructor Overloading in java?

→ In the same class if we write more than one constructor where parameter must be different then it is called Constructor overloading.

55)What is instance block?

→ It is a special block in java which is automatically executed at the time of creating the object.

56)What is blank final field in java?

→ If a final instance variable is not initialized at the time of declaration , then it is called blank final field.

57)What is Inheritance in java?

→ Deriving a new class from existing class in such a way that the new class will acquire the properties and Features in the existing class is called inheritance.

58)Which are the types of Inheritance in java?

- i) Single level Inheritance
- ii) Multi level Inheritance
- iii) Hierarchical Inheritance
- iv) Multiple Inheritance
- v) Hybrid Inheritance

59)How many ways we can use Super keyword?

→ i) To access Super class variable

ii) To access Super class method

iii) To access Super class constructor

60) What is Variable Hiding in java?

→ Whenever Super class variable name and Sub-class variable name both are same then it is called Variable Hiding.

61) Why java does not support multiple Inheritance?

→ -Multiple inheritance is a situation where a sub class wants to inherit the properties of two or more than Two super class.

-In every constructor we have Super() or this() . when compiler will add Super() to the first line of constructor then we have an ambiguity issues that Super() will call which Super class constructor.

-that's reason java does not support multiple inheritance.

62) What is Access modifier in java?

→ -In order to define the accessibility level of the class as well as member of the class we have 4 access Modifier in java.

i) **private**(Within the same class)

ii) **public**(No restriction)

iii) **protected**(Within the same package or even from another package)

iv) **default**(Within the same package)

63) Can we write a java program without main() method?

→ It is possible to write a java program without main method till JDK 1.6 onwards . at the time of loading the .class file JVM will verify the presence of main() in the .class file . If main method is not available then it will Generate a runtime error that “main method not found in so so class”.

64) How many ways we can load the .class file into JVM memory?

→ i) By using java command

ii) By using Constructor

iii) By accessing static data member of the class.

iv) By using inheritance

v) By using Reflection API.

65)What is the difference between java.lang.ClassNotFoundException and java.lang.NoClassDefFoundError?

→-**java.lang.ClassNotFoundException**(It occurs when we try to load the required .class file at RUNTIME by Using Class.forName(String class Name) statement or loadClass() Static of ClassLoader class and If the Required .class file is not available at runtime then we will get an exception i.e java.lang.ClassNotFoundException.

-**java.lang.NoClassDefFoundError**(It occurs when the class was present at the time of COMPILATION but at Runtime the required .class file is not available (manually) deleted by user).

66)What is Static Block in java?

→-It is a special block in java which is automatically executed at the time of loading .class file.

67)What is HAS-A relation?

→If we use any class(Engine class) as a property to the another class(Car class) then it is called HAS-A relation.

68)What is Association in java?

→Association in Java means a relationship between two classes — they are connected but still independent.

69)What is Autoboxing in java?

→When we convert the primitive data types into corresponding wrapper object then it is called AutoBoxing.

70)What is Unboxing in java?

→Converting wrapper object to corresponding primitive type is called Unboxing.

71)Define is Method Overriding in java?

→Writing two or more non-static method in super and sub class in such a way that method name along with method parameter must be same is called Method Overriding.

72)What is upcasting in java?

→It is a process to assign sub class object to the super class reference variable.

73)What is Downcasting in java?

→It is a process to assign sub class object(reference is super type) to sub class reference variable.

74)Define @Override Annotation in java?

→ @Override annotation is metadata and its optional but its always a good practice to write @Override annotation before the overridden method so compiler as well as user get the confirmation that the method is overridden method and it is available in the super class.

75)Can we override private method?

→ No, We can't override private method because private method are not visible (not available) to the sub class hence we can't override.

76)Role of access modifier while overloading a method.

→ While overriding the method from super class , the access modifier of sub class method must be greater or equal in comparison to access modifier of super class method otherwise we will get compilation error.

77)Define instanceof Operator in java?

→ It is used to verify whether a reference variable is pointing to a particular type of object or not?

78)What is the difference between new keyword and newInstance() method?

→ -new keyword is not suitable to create the object for the classes which are coming from database or files at runtime dynamically.

-In order to create the object for classes which are coming at runtime from database or files , we should use newInstance() method.

79)Define final keyword in java?

→ It is used to provide some kind of restriction in our program.

- i) **To declare a class as a final** (Inheritance is not possible)
- ii) **To declare a method as a final** (Overriding is not possible)
- iii) **To declare a Variable(field) as a final** (Re-assignment is not possible)

80)What is Sealed keyword in java?

→ It is one kind of restriction that describes which classes and interface can extend or implemented from Sealed class or interface.

81)What is Abstraction in java?

→ Showing the essential details without showing the background details is called abstraction.

-In order to achieve abstraction we use the following two concept:

i)**Abstract class** (We can achieve 0-100% abstraction)

ii)**interface** (We can achieve 100% abstraction)

82)Define Abstract class?

→ A class that does not provide complete implementation is defined as an abstract class.

83)Define Abstract method?

→ An abstract method is a method without a body , it only has a declaration, not implementation.

84)What is Anonymous inner class?

→ If we define a class inside a method body without any name then it is called Anonymous inner class.

85)Define interface in java?

→ An interface is a keyword in java which is similar to a class which defines working functionality of class.

86)What is Loose Coupling?

→ If the degree of dependency from one class object to another class is very low then it is called loose coupling.

87)What is Tightly Coupling?

→ If the degree of dependency from one class object to another class is very high then it is called tightly coupling.

88)What is Compile time constant?

→ A Compile time constant is a constant that is evaluated and replace with its value at compile time rather than runtime.

89)Multiple Inheritance by using interface.

→ In a class we have a constructor so , it is providing ambiguity issue but inside an interface we don't have Constructor so multiple inheritance is possible using interface.

90)What is default method in java?

→ This default method provides “default implementation” so the implementer class can override to provide

specific implementation in the class.

91)What is functional interface?

- ➔-A functional interface is an interface which contain exactly one abstract method.
- A functional interface can be annotated be @Functional Interface annotation.

92)What is Lambda Expression in java?

- ➔It is used to concise our code as well as we can remove boilerplate code.

93)What is type parameter<T> in java?

- ➔It is a technique through which we can make our application is independent of data type . It is represented by <T>.

94)Define Predicate<T> functional interface?

- ➔-It is a predicate functional interface available in java.util.function sub package.
- It contains an abstract method test() which takes type parameter<T> and returns boolean value . The main purpose of this interface to test one argument boolean expression.

95)Define Consumer<T> functional interface?

- ➔-It is a predefined functional interface available in java.util.function sub package.
- It contains an abstract method accept() and returns nothing . It is used to accept the parameter value or consume the value.

96)Define Function<T,R> functional interface?

- ➔ -It is a predefined functional interface available in java.util.function sub package.
- It provides an abstract method apply that accept one argument(T) and produces a result(R).

97)Define Supplier<T> functional interface?

- ➔-It is a predefined functional interface available in java.util.function sub package.
- It provides an abstract method get() which does not take any argument but produces/supply/return a value of type T.

98)Define UnaryOperator<T>?

- ➔-It is a predefined functional interface available in java.util.function sub package.

-It is a functional interface in java that represents an operation on a single operand that produces a result of the same type as its operand. This is a specialization of function for the case where the operand and result are of same type.

99) Define BinaryOperator<T>?

- It is a predefined functional interface available in `java.util.function` sub packages.
- It is a functional interface in java that represents an operation upon two operands of the same type , producing a result of the same type as the operands.

100) What is Marker interface?

- If an interface does not contain any field or method , Basically it is an empty interface then it is called Marker interface.

EXCEPTION HANDLING

101) What is Exception Handling in java?

- An exception is an abnormal situation or unexpected situation in a normal execution flow.
- Exception occurs due to the following reason.
- i) The wrong input given by the user.
 - ii) Due to dependency, when one part of the program is dependent on another part to complete the task then there may be a chance of getting an exception.

102) Different Criteria of Exception.

- i) `java.lang.ArithmaticException`
- ii) `java.lang.ArrayIndexOutOfBoundsException`
- iii) `java.lang.StringIndexOutOfBoundsException`
- iv) `java.lang.NegativeSizeArrayException`
- v) `java.lang.NullPointerException`
- vi) `java.lang.NumberFormatException`
- vii) `java.util.InputMismatchException`

103) What is enum in java?

- If we want to declare a Universal Constants then we should use enum concept, all the universal constants must be separated by comma and at the end ; is optional.

104)When we use try block?

→ Whenever our statements is error suspecting statement or risky statement then we should write the Statement inside the try block.

105)When we use catch block?

→ The main purpose of catch block to handle the exception which is thrown by try block. Catch block will only Executed if there is an exception in the try block.

106)When we used finally block?

→ The finally block is used to write code that must be executed, even if an exception is thrown or not.

107)What is Nested try block?

→ If we write try block inside another try block then it is called Nested try block.

108)Difference between Checked Exception and UnChecked Exception.

→-Checked Exception:

- i) Checked by the compiler at compile time.
- ii) You must handle it using try-catch or throws.
- iii) Usually occurs due to external factors (like file not found, database error).
- iv) Must handle it or the compiler shows an error.

-Unchecked Exception

- i) Not checked by the compiler at compile-time.
- ii) These are runtime errors, mostly caused by programmer's mistake (like dividing by zero, null pointer).
- iii) Handling is optional, but good practice.
- iv) No compile error, but crashes at runtime.

109)define throws keyword?

→ The throws keyword is used in Java to declare exceptions. It tells the compiler that a method might throw an exception, and whenever calls that method must handle it.

110)What is Exception Propagation in java?

→ Whenever we call a method and if the callee method contains any kind of exception (checked or unchecked) and if callee method doesn't contain any kind of exception handling mechanism (try-catch or throws) then

jvm will propagate the exception to caller method for handling purpose. This is called Exception Propagation.

111)What is difference between throw and throws?

→ -throw(Throwing the exception object explicitly)

We should use throw keyword to throw the exception object explicitly, In case of try-block, try-block is responsible to create the exception object with JVM as well as throw the exception object to the nearest catch-block, but if a developer want's to throw exception object explicitly then we should use throw keyword.

```
-throw new ArithmeticException();
```

```
-throw new LowBalanceException();
```

After using throw keyword the control will transfer to the nearest catch block so after throw keyword Statements, the remaining statements are un-reachable.

-throws

throws keyword describes that the method might throw an Exception, It also might not, It is used only at the end of a method declaration to indicate what exception it supports or what type of Exception it might throw.

It is used to skip from the current situation so now the exception will be propagated to the caller method or JVM for handling purpose.

It is mainly used to work with Checked Exception.

112)How to develop User-defined Exception?

→ If we want to develop checked then our user-defined class must extends from java.lang.Exception, on the Other hand if we want to develop un-checked exception then our user-defined class must extends from java.lang.RuntimeException.

MULTITHREADING

113)What is Multitasking in java?

→ In multitasking multiple tasks can concurrently work with CPU so, our task will completed as soon as possible.

114)Types of Multitasking.

→ Multitasking is further divided into two categories.

a)Process based Multitasking.

b) Thread based Multitasking.

115) What is Process based Multitasking?

→ If a CPU is switching from one subtask(Thread) of one process to another process then it is called Process based Multitasking.

116) What is Thread based Multitasking?

→ If a CPU is switching from one subtask(Thread) to another subtask within the same process then it is called Thread based Multitasking.

117) What is Thread in java?

→ A thread is a light weight process and it is basic unit of CPU which can run concurrently with another within the same context(process).

118) How to create a User-defined thread?

→ In order to create user-defined thread we can use the following two packages.

- i)java.lang.package
- ii)java.util.concurrent sub package

119) When we used Thread.sleep(long millisecond)?

→ If we want to put a thread into temporarily waiting state then we should use sleep() method.

120) If we write Thread.sleep(1000) then exactly after 1 sec the Thread will re-start?

→ No, we can't say that the Thread will directly move from waiting state to Running state.

The thread will definitely wait for 1 sec in the waiting state and then again it will re-enter into runnable State which is controlled by Thread Scheduler so we can't say that the thread will re-start just after 1 sec.

121) When we used join() method of Thread class?

→ The main purpose of join() method is to put the current thread into waiting state until the other thread finishes its execution.

→ Here the currently executing thread stops its execution and the thread goes into the waiting state. The thread goes into the waiting state. The current thread remains in the wait state until the thread on which the join method is invoked has achieved its dead state.

122)Define the Limitation of Multithreading?

- In Data Race or Race Condition, all the thread try to access the resource at the same time so the result may be corrupted.
- In multithreading if we want to perform read operation and data is not updatable then multithreading is good but if the data(modified data) then multithreading may produce some wrong result or wrong data.

123)What is Synchronization?

- Synchronization makes sure that only one thread can access a critical section of code at a time.

124)Define Method level Synchronization?

- In method level synchronization, the entire method gets synchronized so all the thread will wait at method Level and one thread will enter inside synchronized area.

125)Define Block level Synchronization?

- Block-level synchronization means you lock only a specific part (block) of code inside a method, instead of locking the whole method.

COLLECTION FRAMEWORK

126)What is Collection Framework in java?

- The Collection Framework in Java is a set of classes and interfaces that help you store, manage, and process groups of objects (called collections) easily and efficiently.
 - It provides the following sub interfaces
 - i)**List** (Accept duplicate elements)
 - ii)**Set** (Not accepting duplicate elements)
 - iii)**Queue** (Storing and fetching the elements based on the same order i.e FIFO)

127)Define the Methods of Collection Interface?

- a)**public boolean add(E element)**:- It is used to add an item/element in the collection.
- b)**public boolean addAll(Collection c)**:- It is used to insert the specified collection elements in the existing collection (for merging the collection)
- c)**public boolean retainAll(Collection c)**:- It is used to retain all the elements from the existing elements (common elements)
- d)**public boolean removeAll(Collection c)**:- It is used to delete all the elements from the existing collection.

- e)**public boolean remove(Object element)**:- It is used to delete an element from the collection based on the object.
- f)**public int size()**:- It is used to find out the size of the Collection (Total number of elements available).
- g)**public void clear()**:- It is used to clear all the elements at once from the collection.

128) Define List interface?

- List interface is a sub-interface of collection in java.util package available from jdk 1.2v.
- List interface internally uses array concept so all the elements will be stored based on the index.
- Duplicates are allowed.
- null, homogeneous, heterogeneous values are allowed.
- We can perform automatic sorting by using Collection.sort(List list) because sort() accept List as a parameter.
- It stores the elements on the basis of index because internally it is using array concept.
- In list interface few classes are dynamically growable like Vector and Array List.

129) Define Methods of List interface?

- a)**public boolean isEmpty()**:- Verify whether List is empty or not.
- b)**public void clear()**:- Will clear all the elements, Basically List will become empty.
- c)**public int size()**:- To get the size of the collection's (Total number of elements are available in the collection).
- d)**public void add(int index , Object o)**:- Insert the elements based on the index position)
- e)**public boolean addAll(int index , Collection c)**:- Insert the collection based on the index position.
- f)**public Object get(int index)**:- To retrieve the elements based on the index position.
- g)**public Object remove(int index)**:- Remove the elements based on the index position.
- h)**public void set(int index , Object o)**:- To override or replace the existing elements based on the index Position).
- i)**public boolean remove(Object element)**:- Remove the elements based on the object element. It is the Collection interface method extended by List interface.
- j)**public int indexOf()**:- Index position of the elements.
- k)**public Iterator iterator()**:- To fetch or iterate or retrieve the elements from the collection in forward direction only.
- l)**public ListIterator listIterator()**:- To fetch or iterate or retrieve the elements from the collection in Forward and backward direction.

130)How many ways we can fetch the Collection Object?

→-1)By using **toString()** method of respective class [JDK 1.0]

-2)By using ordinary for loop [JDK 1.0]

-3)By using **forEach()** loop [JDK 1.5]

-4)By using **Enumeration interface** [JDK 1.2]

-5)By using **Iterator interface** [JDK 1.2]

-6)By using **ListIterator interface** [JDK 1.2]

-7)By using **SplitIterator interface** [JDK 1.8]

-8)By using **forEach() method** [JDK 1.8]

-9)By using **Method Reference** [JDK 1.8]

131)Define Enumeration Interface?

→We can use Enumeration interface to fetch and retrieve the objects one by one from the collection because

It is a cursor.

132)Enumeration interface contains two method.

→i)**public boolean hasMoreElements():-** It will return true if the collection is having more elements.

ii)**public Object nextElement():-** It will return collection object so return type is object and move the cursor to the next line.

133)Define Iterator Interface?

→It is used to fetch/retrieve the elements from the collection in forward direction only because it is a cursor.

134)Iterator interface has provided two methods.

→i)**public boolean hasNext():-** It will verify, the element is available in the next position or not, if available it Will return true otherwise it will return false.

ii)**public Object next():-** It will return the collection object and move the cursor to the element object.

135)Define ListIterator Interface?

→It is used to retrieve the collection object in both the direction i.e in forward direction as well as backward direction.

136)ListIterator interface has provided four methods.

→i)**public boolean hasNext():-** It will verify the element is available in the next position or not, if available it

Will return true onwards otherwise it will return false.

ii)public Object next():- It will return the next position collection object.

iii)public boolean hasPrevious():- It will verify the element is available in the previous position or not, if available it will return true otherwise it will return false.

iv)public Object previous():- It will return the previous position collection object.

137)Define Spliterator Interface?

→-It is a cursor through which we can fetch the elements from the collection [collection, array, stream], It is combination of hasNext() and next() method.

-It is using forEachRemaining(Consumer<T> cons) method for fetching the elements.

138)Define the List interface specific classes:

→ i)Vector<E>:- Vector is a special type of list in Java that can grow and shrink automatically.

ii)Stack<E>:- Stack in Collection is a special type of list where you can only add or remove items from the top. It works on a rule called LIFO — Last In, First Out.

iii)ArrayList<E>:- ArrayList in Collection is a resizable array in Java. It stores a group of objects and grows automatically when you add more items.

iv)LinkedList<E>:- LinkedList in Collection is a special type of list where elements are connected like a chain. Each element knows where the next (and previous) element is.

139)What is Fail Fast Iterator in Collection?

→ A Fail-Fast Iterator is a type of iterator in Java that throws an error immediately if you change the collection (like add or remove elements) while you are looping through it.

140)Methods of LinkedList class.

→-i) void addFirst(Object o)

-ii)void addLast(Object o)

-iii)Object getFirst()

-iv)Object GetLast()

-v)Object removeFirst()

-vi)Object removeLast()

141)Define Set Interface?

→ The Set interface in Java is part of the Java Collections Framework and represents a collection of unique

elements.

142)What is hashing algorithm?

→ A hashing algorithm is a special function that converts data (like a string, number, or file) into a fixed-size string of numbers and letters — called a hash code or hash value.

143)Define Comparable<T> and Comparator<T> interface?

-Comparable<T> interface

The Comparable<T> interface is used to define a natural ordering of objects. It allows objects of a class to be compared to each other. When a class implements Comparable, you can compare its objects directly for sorting or other purposes.

-Comparator<T> interface

The Comparator<T> interface is used when you need to define custom ordering of objects. Unlike Comparable, it is not defined inside the class. It's used externally when you want to provide different ways of comparing objects.

144)Define Map(K,V) Interface?

→ The Map interface in Java is part of the Collections Framework and is used to store key-value pairs. It is not a part of the Collection interface but is still considered a part of the Java Collections Framework.

145)Methods of Map<K,V> Interface.

→ i) put(K key, V value)

- Adds a key-value pair to the map.
- If the key already exists, it updates the value.

ii) get(Object key)

- Finds and returns the value for the given key.
- Returns null if the key is not found.

iii) remove(Object key)

- Deletes the key and its associated value from the map.

iv) containsKey(Object key)

- Checks if a key exists in the map.

v) containsValue(Object value)

- Checks if a value exists in the map.

vi) size()

- Gives the number of key-value pairs in the map.

vii) isEmpty()

- Checks if the map is empty (no entries).

viii) clear()

- Removes everything from the map.

ix) keySet()

- Returns a set of all the keys in the map.

x) values()

- Returns a collection of all the values in the map.

xi) entrySet()

- Returns a set of key-value pairs (entries) in the map.
- Each entry contains both the key and the value.

146) Define HashMap<K,V>?

→ HashMap<K, V> is a class in Java that stores data in key-value pairs. It allows you to quickly find, add, update, or remove values by using their keys. Keys must be unique, but values can be duplicated. HashMap does not maintain any order of its elements.

147) Define LinkedHashMap<K , V>?

→ LinkedHashMap<K, V> is a class in Java that stores key-value pairs just like HashMap, but it remembers the order in which you added the entries. So, when you print or loop through it, you see the data in the same order you inserted.

148) Define Hashtable<K ,V>?

→ Hashtable<K, V> is a class in Java that stores key-value pairs like a HashMap, but it is synchronized, meaning it is safe to use in multi-threaded programs. It does not allow any null key or null value.

149) Define WeakHashMap<K ,V>?

→ WeakHashMap<K, V> is a special type of map in Java where the keys are weakly referenced. If a key is no longer used anywhere else in the program, Java's Garbage Collector can automatically remove that key-value pair from the map.

150) Define SortedMap<K,V>?

→ SortedMap<K, V> is a special type of Map where the keys are always kept in sorted (ascending) order.

It automatically sorts the keys when you add them.

151)Define TreeMap<K,V>?

→ TreeMap<K, V> is a class in Java that stores key-value pairs and automatically keeps the keys in sorted (ascending) order. It is part of the SortedMap family.

152)What is Generics?

→ Generics means writing code that can work with any data type (like Integer, String, etc.) without rewriting the code again and again. without rewriting the code again and again.

