**21/6/2023**

**Q. Why the main method is public in java?**

\*In java main method bound to be public, if the main method is public then it is able to invoke by JVM but if the main method is not public then JVM is unable to call it.

JBM calls the main method.

In java if the main method is not public, then the program compiles successfully but the program terminates the run time.

Q. What is static in java?

\*In java static is a keyword or it is a modifier.

In java there are 8 modifiers are available.

1. Static
2. Final
3. Abstract
4. Native
5. Transient
6. Volatile
7. Synchronized
8. Stricttp

Modifiers are implemented either in a class or on a method or in variable. By implementing the modifiers on a class or on a method and on a variable. Programmer imposes some restrictions on a class or on a method or on a variable.

Main method is a static in java.

Java popularly usages 2 types of method. One is Static method or class method, second one is non static method or object method.

Static method called outside the class through the class name where as the non static method called outside the class through an object.

Q. Why the main method is static in java?

\*In java as the main method is static it is treated as a class method ,as it is a class method JVM call it by class name without creating a instance of the class. (Instance means object)

Q. What is void?

\* Void is a keyword in java. Here the return type of the main method is void. As the return type of a main method is void it never return a value to the caller or JVM.

Q. What is Main in java?

Main is the method which is declared by java defined by programmer and call by jVM.

Array is a class in java[].

String is a predefined class in java. Main method having only one argument which is array reference of string type.

Q. What is Reference in java?

\*In java reference behave like a variable. In java variable of a class type is treated as reference.

JVM initializes reference.

In this above example Rashmi is the reference of Demo class, which is not initialized in the first line .IN the second line Rashmi is also reference which is initialized by JVM after I construct a object an object of the demo class.

AS rasmi reference of a demo class it must hold the base address of the demo class object.

Q. How many types of Objects available in java?

There are 2 types of objects available in java.

1) First one referenced object

In java referenced object is powerful because of to that extend object having a reference JVM unable to flash at object.

2) Orphan object or unreferenced object.

This object is treated as unreferenced object unused object or it is called orphan object.

This object is not powerful object, this object is flased by JVM.

Q. What is Call by value in java?

22/6/23

The variable which is declared inside a method

Local variables are bound to be initialized.

Local variables always use no access (default) access specifier.

Local variables allocate memory from stack area.

The variable which is declared outside a method, constructor, parameter and block are treated as instance variables in java.

If we not initialized instance variable the instance variable are auto initialized with default value at the object creation time.

Instance variable allocate memory from the heap area.

Static variable: The variable declared inside a class outside of the method, parameter, constructor and block but having the static modifier is treated as static variable in java.

If we not initialized the static variable then it is auto initialized with its default value at its class loading time.

The static variables allocate memory from method area.

**Method call:**

Method calling in case of Non Static method: One non static method calls another non static method inside the same class either by object or directly.

One non static method calls another non static method outside the class by an object.

One static method calls another static method inside the same class by an object class name directly.

One non static method calls another static method outside the class either by object or by class name.

**Method calling in case of static method:**

One static method calls another non static method inside the same class by an object.

One static method calls another non static method outside the class by the help of object.

One static method calls another static method inside the same class either by object or class name or directly.

One static method calls another static method outside the class either by the object or by the class name.

**23/6/23**

Java Virtual machine: The machine which have no physical existence but work like a physical machine that is called virtual machine.

Virtual machines are two types.

1. Hardware based

2. Software based

\*Hardware based: it provides several logical systems on the same computer on the same device which strongly isolation with the physical machine. We can define logical multiple number of logical machines in the same system.

\*Software based virtual machine: This type of virtual machine act as a run time engine to run a particular program.

JVM: java virtual machine

PVM: python virtual machine.

CLR: Common Language Runtime.

Class Loader: It is the sub system of JVM. It loads the .class file in the JVM occupied memory space.

If you want to design a customized class loader then you have to extend your class from java.lang.class loader.

In this class loader it having 3 phases.

1. Loading

2. Linking

3. Initialization

\*Loading: In this phase class loader read the .class file and load the corresponding data in the method are.

For each .class file JVM will store this information in the method area.

For each .class file JVM store the information these are

1. Fully qualified class name.

2. Fully qualified class name of the parent class.

3. Variable Information.

4. Modifier Information

5. Method information etc.

After loading the information JVM implicitly (automatically) creates an object for every .class file on the heap area. That object will be used by a programmer to get the all the information like method information, variable information etc.

\*linking: In this phase it having 3 phases.

1. Verify

2. Prepare

3. Resolve

\* Verify: In this process of insuring that the binary structure of represent of the class the structural correct or not weather the .class file is generated by the valid compiler or not.

\*Prepare: In this process it allocate the memory static variable and initialize the default value and static blocks are start to executed.

\*Resolve: In this process JVM replace the symbolic name in our program with the original memory reference from the method area.

Initialization Phase: In this phase static variables are initialized by user defined value.

\*JVM Memory Space: In the logical memory component present inside the JVM memory space. Class loader lodes all the information of the .class file in the method area.

The size of method area in not fixed which means the size of application is increased then the size of method area is increased.

When we define static blocks they are stored in method area. When we declare static variable they are stored in method area.

\*Stack Area: The logical memory component present inside the JVM memory space

When we declare a method it will store in stack memory area. When we declare a local variable, local reference it will store in stack memory area.

The size of stack is fixed. If we want to increase the size of the stack area then the program terminate at the run time generated stack overflow error.

\*Heap Memory Area: It is the logical memory component present inside the JVM memory space.

In this phase object variable which means instance variable and object are stored in heap memory area and the heap memory array are stored.

In the heap area memory size is not fixed.

\*Native Method area: it is the logical memory component present inside the JVM memory space. In this area it holds the native code. When we define something in java using native modifier or keyword it will store or occupy in the native method area.

\* Execution Engine: it is responsible for execute the code line by line. If the execution engine found the native code then it transfer into JNI(Java Native Interface).

It is the bridge between execution engine and native library. When it find any native code then it will be checked in the native library if the native code is valid then the JNI execute the code and send back to the e execution engine.

\*Native Library: it is the library which contains c/c++ programming language file ,when java try to communicate with c\c++ then the complier helps the native library .

In the interpretation phase (line by line execution) time execution engine convert the byte code to machine code by the JIT (Just in time) compiler.

Once the machine generates the machine code then the micro processer generate the output.

**24/6/23**

Taking the input from keyboard.

In java we use lot of techniques to take the input from the user by the keyboard.

1. Scanner class

2. Data Input Stream class

3. Buffer Reader class

Scanner class:

IT is a predefined class present in java.util package. The main task is of scanner class it takes the input from the keyboard in binary from.

Scanner class having some predefined method through which programmer takes the input from the keyboard.

Scanner class used to read the statement from the various sources like standard input device, string and file.

Constructors: Scanner class Constructors is overloaded.

Public Scanner (file F)

This con is used to scan or read the statement from a particular file.

Public String (file S)

This Constructors is used to scan or read the data from a particular string.

Public Scanner(Input Steam is)

This Constructors is used to scan or read the data from the standard input device.

When we call a constructors then we have to write Scanner sc = new Scanner (System.in)

In the above example in is the constant of system class which points to standard input device and in is the reference for Input String class.

Method present in scanner class:

Public String next ( )

This method extract a word from the keyboard or from the console.

Public String nextLine ( )

This method extract a line from the keyboard or the console.

Public int nextInt()

This method is used to extract the integer data from the keyboard or from the console.

Public boolean nextBoolean()

This method extracts the boolean type of data from the keyboard or from the console.

Public byte nextByte()

This method extracts the byte of data from the keyboard or from the console.

Public short nextShort()

This method extracts the short of data from the keyboard or from the console.

Public float nextFloat()

This method extracts the float of data from the keyboard or from the console.

Public long nextLong()

This method extracts the long of data from the keyboard or from the console.

Public double nextDouble()

This method extracts the double of data from the keyboard or from the console.

Public void close()

This method is used to close the resources.

Public boolean hasNext()

This method is used to check the next word is present or not.

Data Input String: It is a predefined class present in java.io package , we construct the object of the data input string to take the input from the keyboard in ascii format(American Standard Code for Information Interchange) .Whatever the input we pass from the keyboard they are store in data input string object.

As a programmer we can extract the data from the data input sting object by the help of predefined methods.

These methods are non static. When the readLine () method is used to take the sting type of data from the keyboard and console and readLine method shows one exception that is IO Exception .

This caller of the method is handle in exception.

@Deprecated – Is an annotation. It is instruct to the compiler to avoid the the deprecated warning message.

25/06/23

Q. what is array in java?

\*Array is a predefined class present in java.lang.reflect package. Array is a final class. As it is a final class it can’t be inherited.

But in technical term array is a data structure which holds element in insertion order.

Array is static in nature. Once we construct the array we can’t change the size of the array.

The biggest advantage in java is array element is either primitive data types or object references.

As the array elements are stored in insertion order index play an important role for storing the elements are retrieving the elements.

Array class having an instance variable through which we can extract the size of the array after the array is constructed.

Restriction: When we retrieve the elements from the array if the array index cross the range then the program must terminate at the run time by generating ArrayIndexOutofBondsException.

In java when we develop an array program we are bound to follow 3 steps.

1. Array deceleration: In java we are bound to declare an array. When we declare an array we never specify the size of an array.

Int arr[];

String str[];

In the above example arr is an array reference of int type so it must hold the base address of an array where the array elements are of int type.

On the other hand srt is an Array reference of string type so it must hold the base address of an array where the array elements are string class type.

2. Array Construction:

In java array are construed in run time.

When the array is construed we are bound to specify the size of the array. By construing an array programmer indicates the JVM how much byte of memory JVM should allocate to an array.

After the array is construed JVM auto initialized the array elements through the default value of the array type.

After the elements are auto initialized JVM assign the base address of the array to array reference.

3. Array initialization:

In java Array initialization means programmer puts the elements in to the array.

Programmer puts the elements to the array through the array index.

Q. When I declared an array

\*Q. Why the method having an argument which is array reference of string type?

Command line argument is a technique through which programmer enter the input from the command line at the time of execution of java program

In java every command line argument is separated from each other through a space.

Whatever the command line argument we enter they are always stored in the array of string type because at the time of execution of java program JVM always construct an array of string type for holding the command line arguments.

As JVM always construct an array of string type for holding the command line argument show the main method argument is always array reference of string type because after constructing JVM assign the base address of the array to the main method argument.

Q. When we are not passing any command line argument is the main method is mandatory?

Ans- Yes, main method is mandatory because when we are not passing any command line argument then it also JVM construct an array where the size is 0.

Q. Can I extract command line argument outside the main method?

Ans- yes it is possible ,because main method argument is a reference.

26/06/23

Java having some drawback:

In java string is predefined class in java. It is not considered as primitive data types. Whatever

In java there are 8 types of data types are available

So in java is not possible primitive data types

27/06/23

System- System is a defined class

Out- constant defined in system in class (output device)

Println()-defined is PrintStream class

Q. How we define a constant in a java?

\*In java constant must have some characteristics.

1. Constant is freely called within the same package and outside the package. (public)

2. Constant must be initialized. (final)

3. Constant value never be modified. (final)

4. Constant must be a local variable. (static)

In java we never declare a constant by const keyword but we declare a constant by combined used 3 keywords.

That is public, static and final.

**Variables are used in java**

\*Variables are temporary memory area where we can store the data. In java there are 6 types if variables are used

1. Local

2. Instance

3. Static

4. Final

5. Transient

6. Volatile

Q. What is local variable?

\*In java when we declare a variable within a construct or within a method then that variable is treated as a local variable.

In local variable never be invoked outside the block. It is only invoked within the method where it is declared.

A local variable and a local reference and parameter of the method and constructor occupy the memory from the stack.

When I defined a method it will occupy the memory from the stack area.

**Restrictions of local variable:**

Local variable bound to be initialized before it is used otherwise it will generate compile time error.

Local variable always declare through the default access specifier.

28/06/23

Q. What is instance variable in java?

\* An instance variable in Java is a non-static variable defined in a class outside any method, constructor, or block.

We can declare the instance through any access specifier.

As the main suggests instance variable it is treated as object variable.

When we construct an object of a class than the size of the object depends on instance variable available in the class.

In java instance variables are initialized by JVM at the object creation time.

If the programmer not initialized the instance variable then they are initialized by JVM through the default value of data type.

Each time we construct a new object JVM auto initialized it through the default value.

Calling of instance variable: A non static method call the instance variable in the same class directly where as a non static method

A static method call the instance variable within the same class and outside of the class by an object of the

Calling of methods in java: java popularly usage 2 Types of methods.

1. Static method or class method

2. Non static or object method

\* Non static Method: A non static method call another non static within the same class directly where as a non static method call another method outside the class by an object of that class.

A non static method calls a static method in the same class directly or by the class name or by object where as a non static method calls a static method outside either by a class name or object.

\*Static Method: A static method calls another static method outside the class either by the class name or by object.

A static method calls a non static method within the same class and outside the class by an object.

1/7/23

Blocks use in java: There are 3 types of blocks used in java.

1. Static Block

2. Non static block

3. Synchronized block

\*Static Block: In java static blocks are known as class block. Class blocks are excuted by the class loader at the class loading time. As the static blocks have no name it is implicitly called

Rashmi sir:

Static variable: In java static variable popularly known as class variable. It is declared with in a class but outside the method constructor through static keyword.

Restrictions: a static variable never be declared with the method and constructor.

In java Static variable is initialized by the class loader at the class loading time. If the programmer not initialized the static variable then at the class loading time class loader initialize it through the default value of the data type.

We can declare the static variable through ant access specifier.

There is only one copy of the static variable.

Calling of static variable: A static method call a static variable with in the class directly by a class name or by object.

Where as a static method call a static variable outside the class either by a class name or by object.

A non static method call the static variable in the same class directly by a class name or by object and a non static method call the static variable outside the class either by a class name by object.

In the application we want to specify static variable is first initialized before main method called by JVM.

There is only one copy of static variable.

**Difference between static variable and Instance variable:**

1. Static variable is treated as class variable where as instance variable treated as object variable.

2. Static variable is initialized by the class loader at the class loading time where as instance is initialized by the JVM at the object creating time.

3. There is only one copy of static variable where as instance variables are initialized by JVM each time I construct a new object.

\*Final Variable:

In java when I declare a variable by final keyword then it is treated as a final variable.

We can declare the instance variable as final and static variable as final.

Final variables are bound to be initialized.

The value of the final variable can’t be changed.

\*Note: As we know static variables and instance variables are auto initialized by class loader and JVM if we declare the static variable and instance variable as final then they are bound to be initialized by programmer otherwise it generates compiler time error.

\* We declare the local variable as final because that value of that variable can’t be changed.

2/7/23

Blocks used in Java:

Java popularly uses 3 types of blocks.

1. Static Block

2. Non static block

3. Text block

Q. What is called as static block?

\* Static block is popularly known as class block in java. Static blocks are executed by class loader at the class loading time. As the static blocks are executed by class loader at the class loading time then the static block is first executed before the main method is executed.

Within a class we can define n number of static blocks. But the static blocks are executed sequentially by the class loader at the class loading time.

Within a class Static block executed only once as we know class loader load the class file into memory only once.

\*Note- As the blocks having no name it is always implicitly called.

Q. Can we call main method in java?

Ans- yes, we can able to call the main method in java.

Q. Can we able to call the main method before it is called by JVM?

Ans- Yes, through the static block.

Non Static Block: It is popularly known as object blocking in java. When the block is defined without using any modifier then that block is treated as non Static block or object block in java.

Non static blocks are executed by JVM each time we construct a new object.

Within a class we can define n number of non static blocks but there are executed by JVM sequentially at the object creation time.

Non static blocks are executed by JVM after the object is constructed before the constructor is executed.

\*Text Block: It is popularly known as string block. String is immutable in java. Immutable means after constructing object of the string class we are not able to change the contents of the string.

3/7/23

Q. Enter a number from the command line and check the number is a strong number or not, perfect number is not , prime number or not ,spy number or not and find out GCD and LCD?

Q. What is auto boxing?

It is a powerful feature used only for wrapper class. This feature introduced in JDK 1.4 version.

In case of auto boxing the variable of a particular data type directly a assigned to the subsequent wrapper class object.

Q. What is unboxing?

\* It is another powerful feature which is only implemented on wrapper class.

Unboxing is the alternate of auto boxing. It is introduced in JDK 1.4 version. In case of unboxing the wrapper class of object assigned to a subsequent data type.

Q. When we print the value of the reference what s print?

\* When we print the value of reference it must implicitly call a method of object class.

Q. What is object class in java?

\* Object is a system defined Inheritance. A java class directly supports Inheritance .In java if a class not inherit from any other classes then that class must inherit from object class.

In simple word object is the base class for every user defined and system defined class.

In opl method overriding is a power full feature. Method overriding is possible is case of inheritance. A base class method is override in the child class, when the base class method is override in the child class then method same must be same, method parameter must be same and return type of the method must be same.

A programmer override the method to change the functionality in the child class and added some new functionality in the child class method.

In case method overriding when we construct the child class object then JVM at the run time bind the override method or child class method.

As JVM bind the method in case of method overriding it is known as late binding.

When we print the reference of a class it must implicitly call to string method of object class. The object class to string method return <class name> @address in string form.

4/7/23

OOPS

Q. What is class?

Ans- Class is logical entity. As it is logical entity it does not occupies any space. A class is a template of an object.

In java object oriented concept everything defined in class and class is a predefined class in java present in java.lang package.

Q. What is Object?

\*in java object is a predefined class in java present in java.lang package. In simple word object class is a base class of every user defined and predefined classes.

Object describes the state and behavior of a class. In java create and construct object by the help of new keyword calling the constructor.

Q. Is it object and instance are same?

\*No, object and instance are not same. Memory allocating for an object with the specific value treated as instance.

Object having 3 characteristics State, behavior and identity.

State- Variables

Behavior is called non static method.

Identity – Address

Identity - The property of objects that distinguish or differentiate from another object.

Behavior – It describe the non static method.

State- It describes the variables or data.

Q. What is Reference?

\* Reference just like a variable but variables hold the data and reference hold the address of an object. When we declare a variable of class type it treated as reference.

As reference can be static instance and local. When the reference holds the null value it does not points to any object but on the hand reference not hold null value it points to an object.

Through the reference programmer points to an object and change the behavior state of a class.

Demo dd;

EXAMPLE: In this first line DD value is null so that DD never point to any object.

Dd = new demo;

In this line dd holds the address of an object so that dd points to an object.

Inheritance – it is a technique child class can invoke all the properties of base class. Java directly supports Inheritance. In java all classes supported single and multilevel inheritance but the other hand java never support multiple inheritance. But interface supports multiple inheritance.

When a class inherit from another class then we have to use extends keyword.

When a class inherit from interface we have to use implement keyword.

When a interface inherit from another interface then we have to use extends keyword.

In java interface cant’ be inherting from class.

\*Note- Through the simple word a child class object directly invoke the member of base class.

Advantages of inheritance:

Code reusability.

Supports polymorphism

In java single and, multilevel inheritance supports but multiple inheritance not supported.

Note: In the simple word when we call the child class constructor then it must be called the base class non parameterized constructor or default constructor for this reason the size of child class object depends on child class member but also depends on base class number.

Q. Is it default constructor and non parameterized is same?

\*No, it s not same. When we define any constructor in our class this constructor are called non parameterized constrictor, but on the other hand we not defining any constructor in our class then JVM creates it and then it is treated as default constructor.

When a class define a non parameterized constructor through public and no access specifier but on the other hand when we declare a class through a public keyword then the default constructor define public keyword.

When we declare a class with no assess specifier then the constructor is defined through no access specifier.

But when the class having parameterized constructor in the base class the child class constructor is unable to call it.

Polymorphism:

5/07/23

Operator used in Java: Operators are the symbols which is perform over variable which is known as operand.

Java supports 3 types of operators.

1. Unary

2. Binary

3. Ternary

\*Unary Operator: In java there are 7 types of unary operator are available .

1. unary+

2.unary-

3. ++ (increment)

4. – (decrement)

5. () (typecasting)

6. ~ (bitwise)

7. ! (boolean )

In case of unary operator we are implementing only one operand.

\*Type casting: In java conversion 2 types.

1. Implicit / auto / widdening

2. Explicit / types casting / narrow

In java conversion and casting not only possible in case of primitive data types, but also conversion and casting is possible in case of object reference.

**Type casting and conversion of primitive data type.**

In java only one data type not supports typecasting and auto conversion which is Boolean data type.

Not: Long support integer literal where as float support floating point literal . In java floating point literal having higher precision over integer literal, so even if long occupy 8 bit of memory and float occupy 4 bit of memory long value directly assign to float.

Q. What is known type casting?

11/7/23

Polymorphisms are two types.

1. Static polymorphism

2. Dynamic Polymorphism

\*Static Polymorphism: when a method invokes the method definition which is bind (mark) at compile time by the compiler at that time compiler decides which is called. And it treated as static polymorphism.

\*Dynamic Polymorphism: When a method invokes the method definition which is bind at compile time and compiler can’t decide which method is call at the run time JVM decide or bind the method.

12\7/23

In case of bitwise and operator both the condition is checked.

But in case of logical and operator the 2nd condition may or may not check.

\*Bit and operator case return type id int or boolean where as logical and operator case return type is boolean type.

Q.WHAT is constructor?

In java we call a constructor for construct an object in java.

We can call the constructor only one once for creating an object.

We can define a constructor through ant access specifer.

When we define a constructor we can’t use any modifier.

Constructor is defined to initialized to object. Constructor is overloaded but constructor can’t be override.

Types of constructor is java: constructor is of two types

1. Non parameterized constructor

2. Parameterized constructor

Q is the default and non parameterized constructors are same?

\*No, the default and non parameterized constructors are not same.

Q. What is a default constructor?

\*When there is no constructor with in a class but the programmer call the non parameterized constructor for creating the object of the class then it is treated as non parameterized constructor.

The default constructor is either defined by public access specifer or default access specifer.

Q. what is non parameterized constructor?

\* As a programmer we have to define the non parameterized constructor. We can define the non parameterized constructor by any access specifer.

**Relationship between objects: Java supports 3 types of relationship between objects.**

1. IS-A (Inheritance)

2. HAS-A (Composition)

3. USES-A (Aggregation)

13/7/23

Non Static method: Non static method is popularly known as object method. When the programmer wants to change the property of the object or state of the object then the programmer define the non static method in java. Through the non static method we can set the state of the object, get the state of the object and modify the state of the object.

Accsers methods are pair of methods. Accessers methods are must be public and non static.

Accessor methods are popularly known as setter and getter methods.

Setter methods are used to write the object state.

Where as a getter methods are used to read the object state.

Static method: In java static method means class method. We define a static method in java to describe the class properties.

Restrictions Of static method:

1. Static methods never use this and super key word.

2. Static can’t be override.

Q. What is factory method?

\*Factory method returns the base address of the same class object where it is defined.

Factory method is 2 types

1. Static factor method

2. Non Static Factory method

Example of static method: String class method

15/07/23

This Keyword:

In java This keyword is a non static reference.

As it is a non static reference it holds the base address of the current object.

Restriction of this keyword: as this is a non static reference it never be implemented within the static method and static block.

Usages of This keyword: When the instance variable name and the parameter name is same then the programmer use This keyword to differentiate between instance variable and parameter.

Restriction: This never be implemented within the static method.

Q. When we print the value of this keyword then what it prints?

\*As this is a non static reference when we print the value of this keyword it must implicitly called to string method of object class. On the other hand If we override the to string method of object class then the JVM called the override method.

2d use of this keyword: In java when the programmer wants to call one constructor into another constructor of the same class explicitly then the programmer use THIS keyword. At the time of calling of constructor this must be the first statement.

17/07/23

Q. What is inheritance?

It is a technique of sharing the properties of base class within the child classes. A java class directly supports the inheritance because in java every class inherent from object class. In java a class supports single inheritance and multi level inheritance. But a java class not support multiple inheritance.

Note: Why the size of the child class object not only depends on the members available on the child class but also the members available in the base class?

It is the rule of the every object oriented language. When we class the child class constructor then it must implicitly call the based class non parameterized constructor or default constructor, so the size of the child class object not only depends on the member available in the child class and also the member available in the base class.

Java supports multiple inheritance through interface.

Restrictions of inheritance:

1. A final class can’t be inherited in java.

2. When the base class having a non parameterized of constructor but that is defined by private keyword then that class can’t be inherited.

3. If the base class having the parameterized constructor than that class can’t be inherited.

4. The last constraints is avoided by super keyword.

Q. what is super keyword in java?

\* In java super is a keyword or it is a non static reference which is implemented in case of inheritance.

Super is always points to current base class.

Super is a reference which points to current base class member available in child class object.

Restriction: As super is a non static reference it never be implemented within the static method.

Usages of Super keyword:

1. When the instance variable of the base class is similar to the instance variable name of the child class then the programmer use the super keyword to differentiate between instance variable of the base class and instance variable of the child class.

18/07/23

In java as a programmer

Restriction:

1. At the time of calling of constructor explicitly super must be the first statement in the child class constructor.

2. When we call the base class constructor explicitly by using super keyword then the non parameterized

into the child class constructor.

Note: Can a programmer call same class and base class constructor explicitly by using this and super keyword within a constructor?

\*No, at the time of calling of constructor this and super must be the first statement. We can’t use simultaneously within the single constructor.

3rd use of super keyword: in java super keyword is used in case of method overriding. When the child class non static method wants to call the base class method which is override in the child class then the super keyword must be used.

19/07/23

Method Override: It is a power full feature in Object oriented concept. Programmer overrides the method in the child class. when we override base class method in the child class at that time the method may and the return type and parameter are same . In simple word the total signature of method is same. A programmer over ride the base class method in the child class adding some new properties in the child class method, but on the other hand the base class method is not override in the child class then the child class object able to call base class method. In simple words by the help of method override programmer redefine the base class method in the child class.

Advantages of Method overriding: It supports dynamic polymorphism / run polymorphism / late binding.

In case of dynamic polymorphism we call the method at that time compiler is not unable to bind the method because the method name and parameter, return type is same so that compiler unable to bind it. In the method overriding after the object is constructed by JVM, JVM bind the method by checking which class object is constructed.

Restrictions: In method overriding if the method is private then the method can’t be override in the child class.

When the base class method is static then the method can’t be override in the child class. When the base class method is final then the method can’t be override in the child class.

When the base class method override in the child class the override method must be same access specifier or least access specifier.

In case of method overriding the return type can be different.

Co variant return type

In case of co variant return type the return type of overrided method is the child class reference.

20/07/23

Annotations: We use @override annotation to instructor the compiler that the child class bound to override the method of the base class.

Co variant return type: It is introduced in JDK 1.4. It is used in case of method overriding. In case of co variant return type the return type of the method can be different in method overriding.

Method overloading:

24/07/23

Method Overloading: it is a technique through which java supports early binding or compiler time polymorphism or static polymorphism.

The method overloading is not applicable in every situation. When the method name is same where as the parameters of methods are different then method overloading is a good choice.

In simple word when the task of the object is same where as the parameters are different then overloading is a good choice.

When the task is different, then the method overloading is not a good choice.

In case of method overloading when the method name is same where as the parameters are different, then the compiler treat them as a different method, so in method overloading when we call the method then compiler bind the method and the method which is bind by the compiler at the compiler time is called by JVM at the run time.

As compiler bind the method in case of method overloading it is treated as early binding or compiler time polymorphism or static polymorphism.

At the time of calling of the method if the parameter is not match with the argument then compiler supports auto conversion. (If it is possible)

Q. What is vararg?

Ans- Vararg is a powerful feature introduced in JDK 1.4. We declare a method with varied number of similar type of argument if there is a change in the number of the argument then it is advisable to overload an method to overcome this situation ,java introduced vararg, here the three dot are called as ellipse conversion internally vararg parameter will be converted into one-dimensional array, Here also the value are different from through index.

Restrictions of Vararg:

1. 3 dots must lies conjunctively.

2. There is no space between 3 dots.

3. After Vararg do not pass any other parameter.

4. Don’t define a method by passing more than one vararg parameter.

25/07/23

Dynamic Method Dispatch: It means method dispatch at the run time. It is only applicable incase of inheritance.

In case of Dynamic Method Dispatch we are constructing the child class object but it refers to the base class reference.

As we know when we construct the child class object it holds the member of the child class and also hold the member of the base class. But when we refer to the base class reference compiler always check the availability of the member in the base class.

When we call a member of a child class object through the base class reference, then the compiler checks the availability of the member in base class. If the member is not available in the base class then compiler generates compiler time error but only in case of method overriding the compiler check the availability of the method in the base class. But after the child class object is constructed JVM bind the override method.

This technique in java is called as Dynamic Method Dispatch.

26/07/23

Inner class: When we declare a class inside another class. The biggest advantage of inner class is , it implicitly supports encapsulation.

An inner class member can directly invoke the member of the base class, but in the other hand the outer class member never invoke the inner class member directly.

Either outer class member invokes the inner class member through the inner class name or through the object of the inner class.

We can declare the inner class in java through any access specifer.

An inner class member is invoked outside the outer class through the outer class member.

As inner class is a class it generates a class file after compilation. But it must be outer class name $ inner class name.

The types of Inner class: Java supports 4 types of inner class.

1. Local inner class

2. Static inner class

3. Non Static inner class

4. Anonymous inner class

1. Local inner class: when we declare a class inside a method or constructor then that class is treated as local inner class.

Local inner class is local for that method or constructor.

In simple word we create an object of local inner class within a method or constructor.

The Restrictions of Local Inner class: Local inner class are always declared by no access access specifiers.

We can only construct the object of local inner class inside that method or constructor where it is declared.

Local inner class inherited from another local inner class which is declared within the same method or constructor.

Abstract class or interface:

Dynamic method dispatch is implemented either on abstract class or interface.

Q. what is interface?

The declaration is similar to class declaration. But the biggest advantage of interface is it is supporting multiple Inheritance.

Interface is implicitly abstract in nature as the interface.

Implicitly abstract class can’t be intanceciate.

Note: The child class object is the object of the interface.

Whatever the method we declare within the interface they are bound to override in the child class because the method are implicitly public and abstract, as we known abstract method are declared within a interface, but they are bound to override in the child class.

Q. what is overhiding?

27/07/23

Package: In java it is similar to a directory. As directory contains sub directories and files package also contain sub packages and sub files.

In technical word package encapsulates class files according to their functionality.

In java the default imported package is java.lang package.

Ex- Java.io package is a system defined package which encapsulate the system defined classes and interfaces.

Package is of two types: 1. System defined package

2. User defined package

System defined: Whatever packages are already defined in java is treated as System defined package. Ex java.lang, java.sql and java.util , java.io

Java. Lang package is

User defined package: When we develop a small application we store all the classes in a single location or single package.

In java when we declare a package at that time we are bound to follow a guideline that the directory name must be similar as my package name.

But when we develop a live project it is always specified that to store the classes and interfaces in different packages.

To develop a package program we are bound to follow 3 steps.

1. Package declaration

2. Import the class

3. Class declaration

Package declaration: We are bound to declare a package in java when we declare the package we are bound to follow a guide line that the package name must be similar to as the directory name.

Import the class: As a programmer we are bound to import the class which is not available inside the java.lang package.

Here import is the keyword that is used to import the package java.util.

Scanner is the predefine class present in java.util \* package.

Here we import the scanner class by \* asterisks symbol. Here the compiler instructs the class loader to load the class file of java.util package which is used in that program.

Class declaration: As we know the class we declared either used the public access sepcifer or we declared the class by default access specifer.

Package avoid conflict and class colisation.

Package encapsulate the class file according to the functionality basics.

10/08/23

Array: It is a predefined class presented in java.lang.reflect.package.

It is a final class. As it is a final class it cant be .but in technical term array is a data structure which stores the similar types of elements. The biggest advantage in java array it either primitive data types or object references. In simple words once we construct the array

Java.lang: It is fundamental package in java. Java compiler implicitly invoke java.lang package.

Java.lang package contains some predefined classes and interfaces.

Array class

String class

StringBuffer class

StringBuler class

Math class

Enum

System class

Runtime class

Process class

Class class

Object class

Reflection

Exception class

Error class

Throwable class

Cloneable class

Runnabe class

Thread class

Array class: it is a pre-defined class present in java.lang.refect.package.

Array is a final class. As it’s a final class it can’t be inhertated.

But in technical term array is a data structure that holds similar type of elements.

Array is static in nature. Once we construct the array means we can’t change the size of the array.

In java array elements either primitive data types or array elements are of object reference.

When we develop an array program we are bound to follow 3 steps.

1. Array declaration

2. Array Construction

3. Array Initialization

Restrictions ; When we retrieve the array elements if the array index cross the range then the program must terminate at the run time by generating ArrayINdexOutofBundException.

\*After the array is constructed if the programmer wants to extract the size of the array then programmer use the instance the variable length variable.

When the programmer retrieve the elements from the array then the programmer use the for loop

\*Array holds the elements in insertion order so in index play an important role for retrieving the array element.

Array declaration: in java Array declaration is mandatory. In java when we declare an array we never specifie thesize of the array.

In java array is a system defined class. So when we declare an array it is called as array reference.

As it is an array reference it must hold the base address of an array.

Array Construction: In java there are different ways are available to construct an array. The easiest way of constructing an array is called as dynamic construction and Initialization of array.

In this case at the array construction time array elements are initialized by the specified value. But the main problem of this technique is here we declare the array construct the array and initialize the array in the same line.

But the main advantage of this technique is its execution is faster and it is simple to develop the array program

2nd technique of array construction: To avoid the drawback the first technique java introduced another technique which can dynamically construct and initialize the array. It is similar to the 1st technique but here the advantage is we can declare the array the array outside the block but construct and initialized the array in some other box.

3rd technique is the popular tech to develop the array program. Here at the time of array construction we are bound to specie the size of the array.

When we specie the size of the array we instruct the JVM how much byte of memory JVM should allocate to an array at the array construction time.

When the array is contracted JVM first auto initialized the array elements through the default value of the arraty type the JVM assign the base address of the array to the array reference.

Array Initialization: It means programmer put the elements into the array. In java programmer put the elements in to the array through the array index.

Index always starts from zero. JVM recognized the appropriate position of the array through in mathematical calculation = base address + (index \* size of the data base).