[04/01, 7:29 pm] +91 99754 04785: **Java installation procedure:**

Step1: Command to check java version: java -version

step2: check operating system: right click on This PC/My Computer--> properties-->Device Sepecifications--> system type

os 32 bit--> 8 (1.8)

os 64 bit--> 11 -16

Step3: how to download java

search download java 1.8-->click on oracle.com--> scroll down upto java 8/11-->windows --> click on 32/64 related file to download java

Step4: install downloaded java file

Step5: set java path:

5A: Copy java path

open C drive-->program files-->java-->jdk/jre(jdk prefered)-->bin folder-->copy bin folder path (control + C)

Step5B: Set java path

right click on This Pc/my computer-->properties-->advanced system setting-->envirnment variable-->

user variable--> check for "path" variable

--Case1: already path variable exist -->click on path variable-->edit-->new-->ctrl+ V(Paste) --> ok--ok

--Case2: no path variable--> click on new --> enter variable name i.e. -"path"--> variable value -ctrl+ V(Paste)--> ok-->ok

step6: verify java installed or not ?

--> check for java version

execute command --> java –version

[04/01, 7:29 pm] +91 99754 04785: **Diff versions eclipse IDE:**

versions:

oxygen,neon,marsh --> java 8

photon(latest) --> java version 11 & above

eclipse installation procedure:

Step1:

-->search download eclipse photon-->click on https://www.eclipse.org/-->scroll down MORE DOWNLOADS-->4.15/4.16-->

scroll down Eclipse IDE for Java and DSL Developers-->Windows x86_64--> eclipse-dsl-2020-12-R-win32-x86_64.zip

Step2:

--open downloads folder --> unzip eclipse file ---> open eclipse folder--> double click on eclipse application file (blue colour icon)-->

it should ask for workspace path--> keep as it is(default path)-->select checkbox-->launch--> welcome page

[05/01, 7:38 pm] +91 99754 04785: **Java 1st program:**

1. Create java project
2. create java package
3. create java class
4. main method
5. printing statemnt
6. save program (Control + s)

7. run program (click green btn)

8. check output--> Console tab

Shortcuts in eclipse:

1. main method: type "main"+control+space

2. Printing statement: type "syso"+control+space

String, System--> Caps

Step1: create java project--> file--> new--> java project--> enter project name--> finish

Step2: create java package--> right click on project name/Src folder-->new--> package--> enter package name--> finish (create module/don't create module--> don't create module)

Step3: create java class-->right click on package name-->new-->class-->enter class name--> finish

Step4: create main method

Step5: create printing statement

1 java project--> multiple packages-->

1 packages--> multiple classes

1 class--> multiple method--> main method

1 main method --> multiple printing statements --> to print messages

[10/01, 7:35 pm] +91 99754 04785:

Data Types:

Data type are used to represent type of data or information which we are going to use in our java program.

In java programming it is mandatory to declare datatype before declaration of variable.

In java datatypes are classified into two types :

1. Primitive datatype.
2. Non-primitive datatype.

1.Primitive datatype:

There are 8 type of primitive datatypes.

all the primitive datatypes are keywords.

* Memory size of primitive datatype are fix.

The types of primitive datatype are:

Note:- keyword starts with lower case

Primitive datatype starts with lower case

syntax: datatype variablename;

1.(Numeric + Non-decimal):-

Ex: 80,85,10,5..etc

	Data Type	Size	
1.	byte	1 byte	
2.	short	2 bytes	
3.	int(imp)4 bytes		
4.	long	8 bytes	l

1GB=1024MB

1MB=1024KB

1KB=1024Byte

1Byte= 8bit

2.(Numeric + decimal):-

Ex: 22.5,22.8,6.4....

5.	float(imp)	4 byte	f
6.	double	8 byte	

3. single Character :-

Ex: A,B,X,Z.

7.	char	2 byte

4.Conditional:-

Ex: true,false.

8.	boolean	1 bit

----

2. Non-primitive datatype:

There are 2 types of non primitive datatypes .

all the Non primitive datatypes are identifiers.

* Memory size of non primitive datatype is not defined or not fix.

Note: Identifier starts with capital letter.

Non-primitive datatype starts with capital letter.

e.g. String, className

[10/01, 7:35 pm] +91 99754 04785: 1.Variables:

Variables are nothing but piece of memory use to store information.

one variable can store 1 information at a time.

Variables also used in information reusability.

To utilise variables in java programing language we need to follow below steps:

1. Variable declaration (Allocating/Reserving memory)
2. Variable Initialization(Assigning or Inserting value)
3. Variable Usage

Note:- According to all programming language dealing with information directly is not a good practice to overcome this variables are introduced.

[17/01, 7:34 pm] +91 99754 04785:\

Methods:

A method is a block of code which only runs when it is called.

Methods are used to perform certain actions, and they are also known as functions.

You can pass data, known as parameters, into a method.

Why use methods? To reuse code: define the code once, and use it many times.

1. main method (pre-defined)

In any Java program, the main() method is the starting point from where compiler starts program execution.

So, the compiler needs to call the main() method.

without main method we can't run any java program.

2. Regular method (user defined)

1. static regular method

1. static regular method call from same class --> methodname();

2. static method call from diffrent/another class -->className.methodname();

2. non- static regular method

3. non-static method call from same class --> 1. create object of same class 2. objectname.methodname();

4. non-static method call from diffrent/another class --> 1. create object of diff class 2. objectname.methodname();

Note: At the time of program execution main method is going to get executed automatically,

where as regular methods are not going to get executed automatically.

At the time of program execution priority is sceduled for main method only.

To call a regular method we need to make call method call from main method,

until unless if the method call is not made regular method will not get executed.

regular methods can be called multiple times.

5. method without/zero parameter

6. method with parameter.

[24/01, 7:38 pm] +91 99754 04785:

Types of variables:

1. local variable:

The variable which is declared inside the method/block/constructor is called local variable.

scope of local variable remains only within the method/block/constructor.

local variable is also called temporary variable.

2. global variable:

the variable which is declared outside the method/block/constructor is called global variable.

scope of global variable remains throught the class.

global variable is also called permanant variable.

3. static/class variable:

1. static variable call from same class -->variableName
2. static variable call from diff class--> className.variableName

Note: we can access static global variable in both static & non-static method

4. non-static/instance variable: (instance-object)

3. non-static variable call from same class
4. non-static variable call from diff class

[25/01, 7:29 pm] +91 99754 04785:

Java Keywords:

Keywords are predefined, reserved words used in Java programming that have special meanings to the compiler. For example:

abstract		assert	boolean		break	byte		
case	catch	char	class	const				
continue		default	do	double	else			
enum	extends	final	finally	float				
for	goto	if	implements		import			
instanceof		int	interface		long	native		
		new	package		private	protected		public
return	short	static	strictfp	super				
switch	synchronized		this	throw	throws			
transient		try	void	volatile	while			

Java identifiers

Identifiers are the name given to variables, classes, methods, package etc

Rules for Naming an Identifier

1. Identifiers cannot be a keyword.
2. Identifiers are case-sensitive.
3. It can have a sequence of letters and digits. However,

   it must begin with a letter, $ or _. The first letter of an identifier cannot be a digit.
4. It's a convention to start an identifier with a letter rather and $ or _.
5. Whitespaces are not allowed.Similarly, you cannot use symbols such as @, #, and so on.

eg.

// int static, String for, class while

s1, m1, str2, sample1234

abc1, $ab1, _s1,

//1abc, 2s, 5c1,

// s 1, str 2, abc 5,

// abc@1, str#2

[27/01, 7:36 pm] +91 99754 04785:

Types of Constructor

1. Default Constructor

2. User defined Constructor

1. Default Constructor

If Constructor is not declared in java class then at the time of compilation compiler will provide Constructor for the class

If programer has declared the constructor in the class then compiler will not provided default Constructor.

The Constructor provided by compiler at the time of compilation is known as Default Constructor

2. User defined Constructor

If programer is declaring constructor in java class then it is considered to be as User defined constructor.

User defined Constructor are classified into 2 types

[28/01, 7:26 pm] +91 99754 04785:

Constructor:

A constructor in Java is a special member of class that is used to initialize objects/variables.

The constructor is called when an object of a class is created.

At the time of constructor declaration below points need to folow:

1. Constructor name should be same as class name
2. you should not declare any return type for the constructor(like void).
3. Any no of constructor can be declared in a java class but constructor name should be same as class name,

   but arguments/parameter should be diffrent--> Constructor overloading.

Use of Constructor

1. To copy/load all members of class into object --> when we create object of class
2. To initialize data member/variable

Types of Constructor

1. Default Constructor

2. User defined Constructor

1. Default Constructor

If Constructor is not declared in java class then at the time of compilation compiler will provide Constructor for the class

If programer has declared the constructor in the class then compiler will not provided default Constructor.

The Constructor provided by compiler at the time of compilation is known as Default Constructor

2. User defined Constructor

If programer is declaring constructor in java class then it is considered to be as User defined constructor.

User defined Constructor are classified into 2 types

1. Without/zero parameter constructor

    // example-without parameter constructor --eg. addition

2. With parameter constructor

    // example-with parameter constructor- only 1 constructor -- eg. addidtion

    // example-with parameter constructor- multiple constructor -- eg. addition

[01/02, 7:37 pm] +91 99754 04785:

Inheritance:

It is one of the Oops principle where one class acquires properties of

another class with the help of 'extends' keywords is called Inheritance.

The class from where properties are acquiring/inheriting is called super/base/parent class.

The class too where properties are inherited/delivered is called sub/child class.

Inheritance takes place between 2 or more than 2 classes.

Inheritance is classified into 4 types:

1. **Singlelevel Inheritance**
2. **Multilevel Inheritance**
3. **Multiple Inheritance**
4. **hirarchicle**

1. Singlelevel Inheritance:

It is a operation where inheritance takes place between 2 classes.

To perform singlelevel inheritance only 2 classes are mandatory.

2. Multilevel Inheritance:

Multilevel Inheritance takes place between 3 or more than 3 classes.

In Multilevel Inheritance 1 sub class acquires properties of another super class &

that class acquires properties of its another super class & phenomenon continuous.

[02/02, 7:25 pm] +91 99754 04785: 4. Hirarchicle Inheritance:

multiple sub classes can acquire properties of 1 super class is known as hirarchicle Inheritance.

This & super Keyword

1.this keyword

this keyword is use to access global variable from same/current class.

2. super keyword

super keyword is use to access global variable from super class.

[03/02, 7:32 pm] +91 99754 04785:

Access specifiers:

Access specifiers are use to represent scope of members of class(variables,methods,constructor).

In java Access specifiers are classified into 4 types

1. private

2. default

3. protected

4. public

1. private: If you declare any member of class as private then scope of that member remains only within the class

It can't be access from other classes.

2. default: If you declare any member of class as default then scope of that member remains only within the package

It can't be access from other packages.

There is no keyword to represent default access specifier.

3. protected: If you declare any member of class as protected then scope of that member remains only within the package

that class which is present outside the package can access it by one condition ie. inheritance operation

4. public: If you declare any member of class as public then scope of that member remains throught the project.

[19:27, 04/02/2022] Automation Sanjay Sir:

Abstract Class:

A class declared with "abstract" keyword is called abstract class.

an Abstract class is nothing but an incomplete class where programer
can declare complete as well as incomplete methods in it.

programer can declare incomplete methods as abstract method, by declaring keyword
called "abstract" infront of method.

we can't create object of abstract class, to create object of abstract class we need
to make use of concrete class.

Concrete class:

A class which provides definations for all the incomplete methods which are
present in abstarct class with the help of extends keywords is called concrete class.

can 1 Abstract Class extends another Abstract Class ? yes

[19:32, 09/02/2022] Automation Sanjay Sir:

diff types of JVM memories:

1. Heap area--> non-static method declaration
2. Static pool area--> static method declaration
3. method area --> static & non-static method defination
4. stack --> main()--> method execution flow

compiletime --->

runtime/executiontime -->

Polymorphism:

It is one of the OOPs principle where one object showing diffrent behaviour at diffrent stages of life cycle.

Polymorphism is an latin word where poly stand for many & morphism stands for forms.

In java Polymorphism is classified into 2 types:

1. Compiletime Polymorphism
2. Runtime Polymorphism

1. Compiletime Polymorphism:

In Compiletime Polymorphism method declaration is going to get binded to its defination at compilation time,

based on argument/input/parameter is known as compiletime Polymorphism.

As binding takes during compilation time only, so it is also known as early binding.

//once binding is done, again rebinding can't be done, so it is called static binding.

Method overloading is an example of compiletime Polymorphism.

2. Runtime Polymorphism:

In Runtime Polymorphism method declaration is going to get binded to its defination at Runtime/execution time,

based on object creation is known as runtime Polymorphism.

As binding takes during Runtime/execution time, so it is also known as late binding.

//once binding is done, again rebinding can be done, so it is called dynamic binding.

Method overriding is an example of Runtime Polymorphism.

Method overloading:

Declaring multiple method with same method name but with diffrent argument/parameter/inputs in a same class is called

method overloading

Method overriding:

Acquiring super class method into sub class with the help of extends keyword & changing implementaion/defination

according to subclass specification is called method overriding

[19:31, 11/02/2022] Automation Sanjay Sir:

Casting:

Converting one type of information into another type is called casting

In java casting is classified into 2 types:

1. primitive casting
2. non-primitive casting

1. Primitive-casting:

Converting one data type of information into another data type is called Primitive-casting

primitive-casting is classified into 3 types:

1. implicit casting

2. explicit casting

3. boolean casting

1. implicit casting:

converting lower data type info into higher data type info is called implicit casting.

implicit casting is also called widning casting, where memory size goes on incresing,

eg.

```
int a=5    // (memory size of int is 4 byte)
sop(a)     // 5
long b = a   //(memory size of long is 8 byte)
sop(b)       //5
```

2. explicit casting:

converting higher data type info into lower data type info is called explicit casting.

explicit casting is also called narrowing casting, where memory size goes on decresing.

In explicit casting data loss takes place

eg.

```
double b = 2.5   //(memory size of double is 8 byte)
```

```
sop(b)        //2.5
float a= (float)b    // (memory size of int 4 byte)
sop(a)         //2.5
```

3. boolean casting:

boolean casting is considered to be incompatible casting type, because boolean data type is unique type of data type

where information is already predeclared inside it.

```
boolean str = true
```

2. non-primitive casting

converting one type of class into another type of class is called non-primitive casting.

non-primitive is classified into 2 types:

** 1. up casting

2. down casting

1. up casting:

Assigning subclass property into superclass is called upcasting.

before performing upcasting 1st we need to perform inheritance operation.

after performing inheritance, the property which are present inside superclass comes into subclass

In the subclass programer can declare new properties.

At the time of upcasting operation the properties which are inherited from superclass are only eligible for the upcasting operation.

The new property which were declared inside subclass are not eligible for upcasting operation.

2. down casting:

Assigning superclass property into subclass is called downcasting.

before performing downcasting 1st we need to perform upcasting.

[19:34, 14/02/2022] Automation Sanjay Sir:

Generalization:

Extracting all the important common properties from sub class & declaring it in super class(ie. super interface) & providing

implementation/defination according to subclass specification is called Generalization.

generalization file can be normal java class or abstract class or Interface, but only Interface is recommonded.

[19:34, 14/02/2022] Automation Sanjay Sir: package generalization;

```
//super interface  or generalization file
public interface SimCard
{

        void sms();
        void audioCalling();
        void internet();

}
```

}[19:29, 15/02/2022] Automation Sanjay Sir:

String class:

1. String is non-primitive data type, memory size is not fixed.
2. String is use to store collection of characters.
3. String is a inbuilt/ready made class present inside "java.lang" package.
4. String class is final class can't be inherited to other classes.
5. At the time of String declaration, initialization, object creation takes place.

6. String objects are immutable in nature/can't be change.

7. Object creation of String can be done in 2 ways:

   1. Without using new keyword

   2. Using new keyword

8. String objects are going to get stored inside String pool area which is present inside heap area.

String pool area:

It is use to store String objects.

It is classified into 2 areas:

1. Constant pool area
2. Non-constant pool area.

1. Constant pool area:

1. During object creation time if you don't make use of new keyword then

   object creation takes place inside constant pool area.

Duplicate objects are not allowded inside constant pool area.

2. non-constant pool area:

2. During object creation time if you make use of new keyword then

   object creation takes place inside non-constant pool area.

Duplicate objects are allowded inside non-constant pool area.

[19:35, 16/02/2022] Automation Sanjay Sir:

String pool area:

It is use to store String objects.

It is classified into 2 areas:

1. Constant pool area
2. Non-constant pool area.

1. Constant pool area:

1. During object creation time if you don't make use of new keyword then

   object creation takes place inside constant pool area.

Duplicate objects are not allowded inside constant pool area.

2. non-constant pool area:

2. During object creation time if you make use of new keyword then

   object creation takes place inside non-constant pool area.

Duplicate objects are allowded inside non-constant pool area.