DURGA SOFT SOLUSTION

Language fundamental:

Identifiers and reseverd keyword:

1.A name in java is consider as identifiers to identify which can be used for identification purpose it can be method name or variable name or class name or lable name

2.A to Z,a to z ,0 to 9,$,\_ These character are only allowed identifier should not start with digth 123total

3.There is no length limit to java identifiers but it is not recomented to lengthy identifiers .We cant use reseeverd words as identifiers int x =10; Int if =20;

4.all predifien class name interface name we can use as identifiers eg int string =888;

System.out.println(String)

Weven though It is acceptable it is not god programming practice the redablity is going to decrease

Reseverd words:

In java some waords are reseverd some meaning are functionality such type words are reseverd words

Total 53: keywords for data types :bite short char int long float double boolen

Keywords for flow control:

If else ifelse switch case break while for continue do default return

Keywords for modifiers:

Public private protected static final abstract synchronice native strickfp tranciant volatile

Keywords for exception handling:

Try catch finally throws throw asserts

Keywords for class

Class interface extends implimants package import

Object related keywords

New instanceof super this

Void return type keyword

Void in java is mandetary if method whont retuen any thing than we have to declare with void

Null default value for object reference

Enam keyword we can use enam to define group of named constancies

Condtional statement: 1.if-else & switch

Control statement : while &for &do- while

If(condition){ //only retarning true or false statement will execute in condition statment

}

Else if(condition) {

Thes will execute if condition is not true

}

Else{

If condtion is false it will automatically execute

When should we go for condition statement and control statement?

If you want to check for certain contation than go for condtion statement

If you want to do repetated task than go for control statement

Control statement:

For(loop startvalue; loop end value; loop incr/decemant value){

}

**for**(x=1;x<=10;x++) {

System.***out***.println(x);

}

While (condition){

}

Int x =1;

While(x<10){

Syso(x);

X++;

}

Import source strucher

There can be any number of class in java

eg:

Class A{

}

Class B{

}

Class C{

}

Class D{

}

Class E{

}

Even though there is many class in what name you will save the file ?

It can be any class untile if there is no public class as well as there should be only one public class name if there is more than one public we will get compalisation error

Coz the file Name and public is only related .no main name is related to

**package** sample;

**public** **class** sample1{

**public** **static** **void** main(String[] args){

System.***out***.println ("hellow world ");

}

}

**class** B{

**public** **static** **void** main (String[] args){

System.***out***.println ("hellow world ");

}

}

**class** C{

**public** **static** **void** main (String[] args){

System.***out***.println ("hellow world ");

}

}

**class** D{

}

you can save any name coz there is no public class if run this code there will be 4 java class after saving it as durga.java

you can run any three class sepratly like class A ,class b, class c ,but you cant run class D coz there is no main class

**Import:**

There are two types of import class are avliable

Explicit and implicit import class

Explicit import means implicit import means

Import java.util.arraylist; import java.util.\*;

In default you should use explicit import because of redablity is going to increase

Two package we no need to import one is java.lang page and other one is default pkg

Eg: if current working directry is present no need to import

A class inside the same directry we no need to import

Eg D: durgaclass /test.java

/sample.java

/selenium.java

Like this there are many class if we want to samle class in selenium class we no need to import

The import concept about import statement

When your using pattern statement in regs pacage :

java

util

regs

pattern

if you want to import pattern you can use import java.\*;

import.java.util.\*;

import java.util.regs.\*;

import java.util.regs.pattern;

you cant use 1 option or 2 becouse package will take . in java it will take all the class but not sub class of util class like wise when your using 2option you can take regs but not take sub class in that

when you want to import statement for sub pakage you have to wright the import statement until sub page level

**Package:**

Package is group macanisam(encapsulation mac) to group related class and interface into single unit is called package

If ther is two date class is there one is in sql and other one is utile so if you want to different them you want to use than package statement come in picture

Advantage of using pack

1.Naming will be very easy

2.Maintanablity

3.Security

4.modularity

Use internet domine name in the reverce to name the package eg: skasc.com use.

Com.skasc.jac

To normaly compile we use javac test.java but in if your using package you should use javac –d . test.java(d represent destinastion) or else we can use full colified name like javac com.skasc.jac.test.java if you don’t use these two it will not run

In java source file only one package can be taken if you take more than one compiler will give left and right

The first non comment statement should be package but not import statement

**Class level modifiers:**

Modifiers defines begavier of the class

You have to give some information to jvm that these class can be accesable by where or not that is called modifires

The applicable modifiers for top level class are:

Public

<default>

Abstract

Final

Strite pf

For inner class modifiers are

Private

Protected

Static

+the top level class modifiers are also applicable

Public class: if class is declared under public any one can access no problem at all

If class is not declared as public it is default

Default class: the default class can be accessable only within the same package not out side the pacakage

**Absaract method:**

Abstract means its incomplit

If you don’t now about implimantasion(means if you don’t now vechile as how many wheels it defance in vechile)you can declar using abstract .abstract method can only declare but not implimant. than you can use abstract the abstract method should ends with ;not with {}. Child class are responsible for implimantion

Eg: class loan

{

Public double getintrset();

}

Which one is valid

Public abstract void m1(){}

Public void m1();

Public abastract void m1();

Public void m1(){}

The first two are not valid coz in 1 abstract must end with ;

The 2 one must end with {} but not with ; coz ; can only used in abstract

The 3, 4 are perfectly valid

If atlest one abstract method is there than 1 abstract class should be used

**Abstract class:**

If class doesn’t have proper implimantion than you can use abstracat class

If you use abstract class no one can use object of the class or cant call the method of the class

When to use abstract class if abstract method or dummy implimantion is used than you can use abstract class

For abstract class object creastion is not possible

**Abstract class vs abstract method :**

If class contain abstract method it must contain abstract class other vise compile time error we will get

Abstract class contain 0 num of abstract class

It can dummy method in it eg:

Public class Test

{

Public void m1(){}

Public void m2(){}

}

Test t=new test();

t.m1(); //this is dummy method

eg2:

abstract class test

{

Public abstract void m1();

Public abstract void m2();

}

Public class subclass extent test

{

Public void m1()

{

}

} // this code contains error coz there are two abstract method so subclass are responasable for the 2 abstract method if you take only 1 method than you will get error

You can solve this problem in 2 ways

1.give m2 method

2.give abstract class for subclass also //this tempravery fixing still you have implimant m2 method in next subclass

**Member modifiers public default:**

**Public member:**

The public can be used any where in the class and outside the class or ouside the package

But you can use public member outside the class or package only if class is also public if class is not public you cant use outside the package

**Default member:**

The default member can be acces with in the class but not out side the class

Package pack1;

Public class A{

Void m1(){

System.out.println(“a is modifiers”)

}

}

Package pack2;

Import pack1.test;

Public class B{

Public staic void main(String[], args){

A surya =new A;

Surya.m1();

}

//we will get eror if you compile this code coz default method or variable can access outside the pacakage

**Private class**

This class means accesable with in the class but outside the class

Data member : variable //highly recommended is private

Methods:public

Protected members:

Protected can me usable only within the package but not outside the pack .in outside pack you can use in child class

And you should use child class reference only but not parent class reperance

Prodected =<defult>+ kid

Eg :

**package** sample;

**class** b{

**protected** **void** m2(){

System.***out***.println("this is protected");

}

}

**public** **class** sample1 **extends** b

{

**public** **static** **void** main(String[] args)

{

b sur=**new** b();

sur.m2();

sample1 su=**new** sample1();

su.m2();

b sury =**new** sample1();//parent reference A child osample1ject sample1

sury.m2();

}

}

This example can use child class reperance but not parent class

Eg:

**package** sample;

**public** **class** sample1{

**protected** **void** m2(){

System.***out***.println("this protected");

}

}

**package** pack2;

**import** sample.sample1;

**public** **class** Test **extends** sample1

{

**public** **static** **void** main(String[] args)

{

Test sur=**new** Test();

sur.m2();

Test su=**new** Test();

su.m2();

Test sury =**new** Test();//parent reference A child osample1ject sample1

sury.m2();

}

}

Conclustion :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Visablity | public | Protected | default | private |
| With in the class | yes | yes | yes | yes |
| From child class of same package | yes | yes | yes | no |
| From non child class of same package | yes | yes | yes | no |
| From child class of outside package | yes | yes | no | no |
| From non child class of outsited package | yes | no | no | no |
|  |  |  |  |  |

Note(advanve java servalates and jsp)

**Interface**

Any service required specification is called interface (srs)

Any contract between client and service provider is interface eg: client came with some of service must required for my to collage to the service provider than it is interface

**How to declar and implimant interface :**

Interface is requirement perpas

Eg :interface inter{

Public void m1();

Public void m2();

}//this is reqirment for interface

Class service implements inter{

Public void m1(){ //first error when ever implimanting an interface method we have to use public its compulsary

}

}

The second error was we have porived implimantion for each and every requirement you solve this is two ways

1.provide implimantion for all requirement

2.declar the class as abstract .subclass are responsible for this

Eg class subservice extent service{

}

**OOPS**

**Data hiding:**

Our internal data should not go out directly than we have to use data hidihg . some sort of validastion should provided

To implimant data hiding you can use private member modifier the main advantage of you using data hiding is security

Eg:

Class account{

Private double balance {

}

Public double getbalance{//validate

If (validate ){

Return balance;

}

Else {

System.out println (“not valid”);

}

}

**Abstraction:**

Ifyou doesn’t have proper implimantion than you can use abstracat

Hiding the internal implemantion and just high light set of services what we are going to offer

Advantage of using abstraction

1.security

2.enhansment(with out affecting end user you can change any internal implimantion and even the technology)

**Encapsulation:**

The proses of grouping set information into a single unit is called encapsulation .Every java class is eg for encapsulation

If any componat is following data hiding and abstraction than it is called encapsulation componat

Encap=data hiding +abstract

Eg;

Class account{

Private double balance(){

}

Public double getbalance (){

//validate

Return balance;

}

Public void set balance (double amount){

//validate

This balance =this.balance +amount;

}

}

//how to find whether ther is encapsulation .if ther is data hiding and abstract

Data hiding =private

Abstract= get method and set method perform in inner but not outer

**Tighly encapsulated :**

If only when every variable is private means private method are tightly encapsulated

If parent class is not tightly enacaspulated no subclass are tightly encapsulation

**Inheritance:**1.is –a relastion ship

2.code reusability //what ever parent as it comes to the child class

3.extented keyword

You cant call child method in parent reperance

Eg:

Class p {

Public void m1(){

System.out.println(“parent”)

}

}

Class c extend p{

Public void m2(){

System.out.println(“child”)

}

}

Class test {

Public static void main (String[] args){

P try =new p();

Try.m1();

Try.m2(); // you cant call m2 method its not in parent class you parent class can only acess parent method

But child can acess parent and child

P try =new c(); //you can creat method using parent reprence and child object

Try.m1();

Try.m2();//you cant call child meyhod using parent repernace

C .c new= C();

c.m1();

c.m2();

C c new=p();//you cant call method using child reperance

c.m1();//compitime error

}

}

Without inheritance

Class house loan{

300 method

}

Class vechile loan{

300 method

}

Class gold loan{

300 method

}

Total 900 method

90hr

With inheritance

Coman method should written in parent

Class paraent {

250 method

}

Class house loan ext**and parent {**

50 method

}

Class vechile extend parent{

50 method

}

Class gold extend parent{

50 method

}

Total= 400 method

40 hrs

Types of inheritance

1.Single inheritance :

Class a

Class b extend a

//1 parent and 1 child class is called single inheritance

2.Multiple inheritance (is also know as parent child relastion but java does not support for this

Class a class b

Class c

3.multilevel inheritance :

This inheritance is also know as grand father father and son relastion ship

Class a

Class b extend a

Class c extend b

4.hirarcial inheritance :

This is the reverse of multiple inheritance

Class a

Class b class c class d

5.hyrbrid inheritance :

Grouping all the inheritance or using all inheritance symentaniysly then it is hybrid inheritance but its not support by java

Multiple inheritance in detail

Why java does not support multiple inheritance

Coz if p1 and p2 are c these are the class

If c call p1 and p2

In p1 there is m1 method

And inp2 there is another m1 method is there

The compiler does not not which m1 method to execut

But in interface concept muptiple inheritance is accepted

Eg :

P1i p2i

Ci

In interface only declarastion but not implimantion is possible

Even ther is multiple declarastion ther is only one implimantion

**Polymorpisam:**

Method signature:

Method name should contiane method name followed by argument types

Eg:

Public int m1(int I, float r)// m1(int float) even the oder should be important int must come first and float should be second

Compiler will use method singnature

Like m1 int 10;

Is present or not if not we will get error

With in the class two method singnataur is not allowed

Eg:

Public class A{

Public void m1(int i){

}

Public int m1(int j){

Rturn 10;

}

}

A now = new A();

Now.m1(10);

Method overloading :

If both method having same name with different argument

Eg m1(int)

m1(long)

case study 1 :

class Test{

public void m1(){

sys.out.println(“no arg method “);

}

Public void m1(int){

Sys.out.print(“int arg method”);

}

Public void m1(double){

Sys.out.print(“double arg method”);

}

Public static void main(string[] args){

Test t= new Test();//is runtime object .method resalustion is taken care by compiler using reperance object

T.m1();

t.m1(10);

t.m1(10.97);

}

Overloading is also no as compile time polymorpishm

Coz new test is run time object

Method resalustion means which method to be executed is method resalustion is taken care by compiler using method reference

Case study 2:

class Test{

Public void m1(int){

Sys.out.print(“int arg method”);

}

Public void m1(float ){

Sys.out.print(“double arg method”);

}

Public static void main(string[] args){

Test t= new Test();

t.m1(10);

t.m1(10.9);

t.m1(‘a’); //if ther is no char method is available than compiler will not produce error instant char automatic promostion to check if there is any int,float,long,double is ther are not. still there is no match than we will get error

}

Case study 3:

Class Test{

Public void m1(object o){

Sopln(“object version “);

}

Public void m1(string){

Sopln(“string type”);

}

Public static void main(string [] args){

Test t= newTest();

t.m1(new object() );

t.m1(“surya”);

t.m1(null);//parent and child are small work we have to go for child and for big work go for parent so string object is going to get the chance

}

}

Case study 4:

Class Test {

Public void m1(string s){

Sopln(“string version);

}

Public void m1(stringbuffer sb){

Sopln(“stringer buffer”);

}

Public static void main(string[] args){

Test t =new Test();

t.m1(“surya”);

t.m1(new stringbuffer(“durga”);

t.m1(null);//we will get error coz string ans string buffer are equal so antiquity problem we will get

}

Case study 5:

Class Test {

Public void m1(int i,float f){

Sopln(“int float version”);

}

Public void m1(float f,int i){

Sopln(“float int version.”);

}

Public static void main(string[] args){

Test t =new Test();

t.m1(10,10.4f);

t.m1(20.4f.20)

t.m1(10,10) //both got matched compiler will give left and right

t.m1(10.4f,34.1f); //both does not got match so we got symbole not found

}

}

Case study 6:

Class animal{

}

Class money extend animal{

}

Class Test{

Public void m1(animal a){

Sopln(animal version);

}

Public void m1(monkey m){

Sopln(“monkey version”);

}

Public static void main(string[] args){

Test t =new Test();

Animal a=new animal();

t.m1(a);

monkey m=new monkey();

t.m1(m);

animal a1=new monkey();

t.m1(a1);

}

}

**Overriding**:

If the parent implimantion is not sataisfied by the child then the child can rewrite the implimantion of the parents

Eg:

Class p{

Public void proopoerty (){

System.out.println(“cash,land “);

}

Public void marry(){ // this is called override method

System.out.println(“applama ,subalama”);

}

Class c extend p{

Public void marry(){

System.oout.printlln(“sunny,miya”);

}

Class Test{

Public static void main(string[] args){

P pa =new P();

Pa.marry();

C ch =new C();

Ch.marry();

P pc =new c();

Pc.marry(); //child method will get chance coz at run time compiler will check child class is overriding or not if it is overriding jvm will give chance to child if not parent will get the chance

}

}

Rules 1: overriding

Covariant means child method name need not to be parent name it can be child name also if objet you can use string name instant.

Eg:

Class A{

Public object m1(){

Return null;

}

Public class B extent A{

Public string m1(){ //it need not to be string it can be string buffer any thing like that

Return null;

}

Rule 2:  
overriding concept is not applicable for private coz child class cant see the parent class if its private but if you try to override in child class it is valid but it is not overriding in indipanted and you cant call the private class

Rule 3:

Overloading is not possible for final method

Rule 4:

Class P{

Public void m1(){

}

}

Class C extend P{

Protected void m1(){ //we will get error coz we cant reduce the scope of the modifier but we can increase the scope

}

}

Rule 5:

You want to now throw and exaption

There 2 exaption uncheaked exception and cheacked exception

The uncheack are run time exception and its child class and error and its child class are unchecked exception

The checked excetion the compiler will check that all exception is under check exception is checked are not

Eg : exception ioexception severlat execption…

If child class metod thrw any checked exception than parent class should thow the same exception or parents

**Overrinding using static method:**

You cant override non static method to static method or static method to non staic method you will get error

But you can use static in both this is not overriding is method hiding if method resolustion is take care compiler yousing reperance operater

Eg:

P p=newP();

P ref=new c ();// in static p will get executed but not run time object C

Overriding with vrg method

Class p{

Public void m1(int…i){

}}

Class c{

Public void m1(int){

}}

Class Test {

P p1=new P();

P1.m1(10);//parent

C p2 =new C();

P2.m1(10);//child

P p3 =new C();

P3.m1(10); // parent// this is not overrinding is overloading so reference type will get the chance

}

Overriding with varirable :

Class p{

String p =”parent”

}

Class c extend p {

String p =”child “

}

Class Test{

Public static void main(string[] args){

P p1= new P();

Sopln(p1.p);

C c1= new c ();

Sopln(c1.p);

P p2 = new c();

Sopln(p2.p);

}

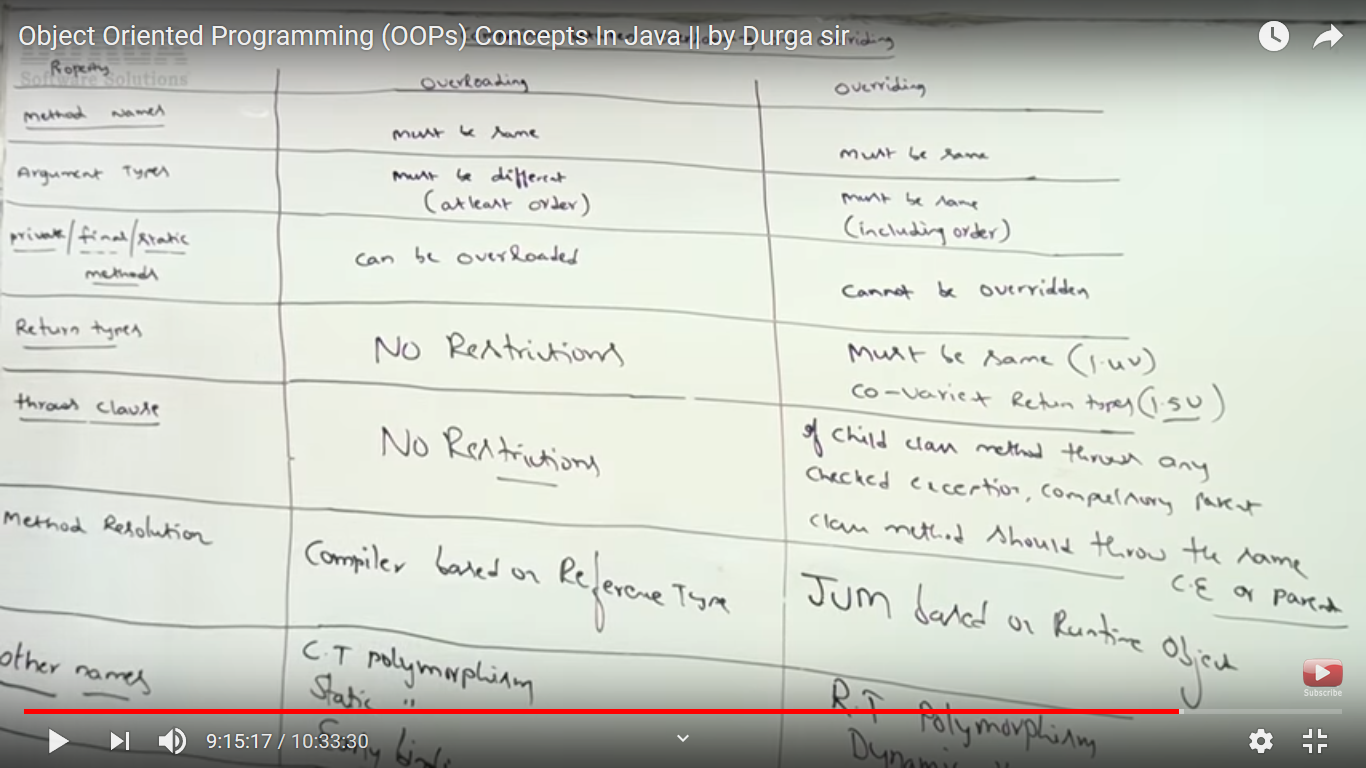
}

verriding concept is only applicable for method not for variable

Variable resalustion is take care by compiler based on reperance variable

These type of concept is called variable hiding or shadowing

Comparistion between overloading and overriding:

****

**Polymorpisam:**

Poly means many

Morpism means form

Object typecasting:

A b=(C) D; object o =new string(“surya”)

Stringbuffer sb= (string buffer)o;

3 rules 1 rule is checked by jvm and 2 by comiler

Comile time check 1:

The type of d and c must have sum relastion either parent to child are child to parent or same relastion

The string buffer and o as parent child relastion

Compile time check 2:

C must be same or it must be child object of A

Jvm error 3:

Stringbuffer must be same or derived type of string

**Constractor:**

**Once the object is created we should inisalise the object that object othe wise it is going to default value**

**How is responsible for insilisation? After creating the object constractor is responsible the constract is reasponsable for inisiation but not creat object.new keyword is responsanble for creating object**

**For every copy of object sprat instance variable is going to creat for every variable jvm is going to provid default value is null for string and 0 for int.**

**this .name=name means current object name**

**for eg:**

**class student{**

**string name;**

**int roll num;**

**student (string name ,int rollnm){//constract name and class name must match**

**this.name=name;**

**this.rollnm=rollnum;**

**}**

**Public staic void main(string[] args){**

**Student s1 =new student(surya 101);**

**}**

**Exception handling**:

Introduction

AN un expected un wanted event that disturm the normal flow of program is called exception

Eg tyer puncher exception

Sleeping exception

File not fount exception

Pupers:

It is highly recommended to handle exception the main objecti ve is grace full terminastion of the program

We should not lose any thing that type is called exception

Having alternative way to continue the reset of the program is nothing but exception handling.exception does mean reparing the exception we need to provide alternative way to continue reset of the program normally

For eg over programming requirement is to read data from remotive file locating at London at run time if our remote file is not available our program should not terminated we have to provide some local file to continue reset of the program normaly this way of definig alternative is nothing but exception handling

Try {

Read file located at londan

}

Catch (filefountexcption ){

Use local file

}

Runtime stack machinsm:

For every thread time jvm is going to creat run time stack each and every method call perpormed my thread will be stored in curresponding stack each entry in the stack is called stack frame or activastion

After compliting method call the corresponding entry will be removed after completing method call the stack will become empty the empty stack will be destroyed by the jvm just before terminating

Class A{

P s v m (string[] args){

dostuf ();

}

P s v m (dostuf){

Domorestuf();

}

P s v m (domorestuf){

Sopln(“hello”);

}