Method overrididng:

Parent class/Super class:

A parent class can have ‘n’ no.of child class.It inherits /extends all properties/methods from parent class.

Child class/sub class

A child class can have one parent class.

e.g.

Rajam



extends



Saraswathi

extends



Jansi



theju Atharva (extends)

Note:

1.theju and atharva extends all properties from jansi,similarly jansi from Saraswathi.

2.theju and atharva share all the methods from jansi,Saraswathi and rajam and it also contains its own methods

3.we cannot access child class methods using parent class object

What is method overriding?

When parent class and child class has the same method name and same no.of arguments between different classes ,then it is called as method overriding.

Pref will be given to child class methods when we execute the prog.

Static polymorphism /compiler -time polymorphism?

Method overriding is known as static polymorphism because many methods with same name and same no.of arguments can be given within different classes and while compiling,it is decided to which method the pref shd be given ,so it is called as compiler -time polymorphism

Dynamic polymorphism /run -time polymorphism?

Child class object can be referred by parent class reference variable.It is also known as top-casting

Only parent class methods and overridden methods can be accessed by dynamic poly and we cannot access child class methods using parent class object

Down casting :

Parent class object cannot be referred by child class reference variable.It can be done by casting option.

Points to remember:

Method overriding is called as static polymorphism.in this case,pref will be given to cgild class overridden method.oly that overridden method will be executed.

If,in case we need to execute parent class overridden method,it is done by dynamic polymorphism or top casting /run-time polymorphism .

i.e.parent class ref variable= child class object

Interface :

It has set of rules which cannot be changed.

1.only method declaration/only method prototype ,no method body inside a interface

2.variables can be declared,but default all the variables are static and final i.e.values cannot be changed in nature.

3.methods are non-static methods by default .

4.no main method inside an interface

5.interface is abstract in nature.i.e.we cannot create object for an interface

Parent interface--jansi



Implements



Theju -child class

1.If a child class implements parent interface,then all the methods in the parent interface shd be defined in the child class and implements keyword shd be used.

2.A child class can implement more than one parent interface.

3.This is calles multiple inheritance because child class can have more than one parent interface .

\*\* interfaces used in selenium:

webdriver driver=new chromedriver();--🡪webdriver-interface

🡪chromedriver()-child class

Webelement element=driver.findelement()

Arraylist Vs hashtable:

Difference between array and arraylist:

Array:

\*it is static,i.e sixe is fixed.we cannot change the size of array

e.g.int i[]=new int[4];🡪 4 values only can be added

arraylist:

\*it is dynamic.i.e size is not fixed.we can add any no.of values and any data type also can be added to the list .

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

ArrayList<Integer> arl=**new** ArrayList <Integer>();

arl.add(100);

arl.add(200);

//arl.add("janu");

//System.out.println(arl.size());

//System.out.println(arl.get(2));

**for**(**int** i=0;i<arl.size();i++)

{

System.***out***.println(arl.get(i));

}

1.arl.size()-size of arraylist-no.of values in the array

2.add(100);values can be added using add method to the arrayliat

3.arr.get(index)-get values by passing index of values

4.to limit the datatype of an arraylist to integer arrayliat i.e.oly integers can be added to the arraylist,

ArrayList<Integer> arl=**new** ArrayList <Integer>();🡪<Integer >can be added

Hashtable:

1.We don’t store values in the index,instead we store them in “key”,”value” pair.

2.h.put() method -🡪used to store keys and values

3.h.get() method🡪is used to get the value of a particular key

4. Hashtable<Integer,String> 🡪we can restrict the data type of both key and value

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

Hashtable h=**new** Hashtable();

h.put(1, 100);

h.put("A", "Jansi");

h.put(2, "Theju");

System.***out***.println("Before ="+h.get("A"));

h.remove(1);

System.***out***.println("after ="+h.size());

Hashtable<Integer,String> h1=**new** Hashtable <Integer,String>();

h1.put(2, "A");

System.***out***.println("after ="+h1.get(2));

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