Agenda:

* Inheritance:

~ Single Inheritance

~ Multilevel Inheritance

~ Hierarchical Inheritance

* Polymorphism

~ Method Overloading

Inheritance:

It is the process by which one class acquires the properties(fields) and functionalities(methods) of another class.

This allows for reusability.

Types of Inheritance:

1. Single Inheritance:

* A child class inherits from a single parent class.

package SingleInheriExample;  
  
public class Parent {  
  
 void showMessage(){  
 System.*out*.println("Parent Class");  
 }  
}

package SingleInheriExample;  
  
public class Child extends Parent{  
 void display(){  
 System.*out*.println("Child class");  
 }  
}

package SingleInheriExample;  
  
public class Main {  
 public static void main(String[] args) {  
 Child obj = new Child();  
  
 obj.display();  
 obj.showMessage();  
 }  
}

o/p:

Child class

Parent Class

Explanation:

A child class inherits the showMessage method from parent class, allowing the access to both parent and child methods.

1. Multilevel Inheritance:

A class is derived from another derived class (forms a chain of inheritance)

package multiLevelInheritance;  
  
public class Grandparent {  
 void message(){  
 System.*out*.println("Grandparent Class");  
 }  
}

package multiLevelInheritance;  
  
public class Parent extends Grandparent{  
 void showMessage(){  
 System.*out*.println("Parent Class");  
 }  
}

package multiLevelInheritance;  
  
public class Child extends Parent {  
 void display(){  
 System.*out*.println("Child Class");  
 }  
}

package multiLevelInheritance;  
  
public class Main {  
 public static void main(String[] args) {  
 Child obj = new Child();  
 obj.message();  
 obj.showMessage();  
 obj.display();  
  
 }  
}

o/p:

Grandparent Class

Parent Class

Child Class

Explanation:

Child class inherits from parent, which in turn inherits from Grandparent, allowing child to access methods from both classes.

1. Hierarchical Inheritance:

Multiple child classes inherit from a single parent class.

package hierarchicalEx;  
  
public class Parent {  
 void showMessage(){  
 System.*out*.println("Parent Class");  
 }  
}

package hierarchicalEx;  
  
public class Child1 extends Parent{  
 void display(){  
 System.*out*.println("Child 1 class");  
 }  
}

package hierarchicalEx;  
  
public class Child2 extends Parent{  
 void display(){  
 System.*out*.println("Child 2 class");  
 }  
}

package hierarchicalEx;  
  
public class Main {  
 public static void main(String[] args) {  
 Child1 obj1 = new Child1();  
 Child2 obj2 = new Child2();  
  
 obj1.showMessage();  
 obj1.display();  
  
 obj2.showMessage();  
 obj2.display();  
 }  
}

o/p:

Parent Class

Child 1 class

Parent Class

Child 2 class

Explanation:

Child1 and Child2 both inherits from parent, due to which gaining the access to its methods.

NOTE: JAVA doesn’t support multiple inheritance (One class inheriting from multiple parent classes\*) to avoid ambiguity.

This can be achieved with the help of interfaces.

Polymorphism:

* Method Overloading(Compile Time Polymorphism)
* A class has multiple methods with the same name but different parameter list.

package calcu;  
  
public class Calculator {  
 int add(int a, int b){  
 return a+b;  
 }  
  
 int add(int a, int b, int c){  
 return a+b+c;  
 }  
}

package calcu;  
  
public class Main {  
 public static void main(String[] args) {  
 Calculator calc= new Calculator();  
 System.*out*.println("Sum of 2 numbers: "+ calc.add(10,80));  
 System.*out*.println("Sum of 3 numbers: "+ calc.add(10,80,90));  
 }  
}

TASK:

Class -> Hotel

Methods -> bookRoom(String roomType){sout(room of \*\*\* type is booked)}

bookroom(String roomType, int days){sout(room of \*\*\* type has been booked for \*\*\* days)}

object creation -> accessing methods

~ Method Overriding

* Super Keyword

~ How to use super keyword?