**Tutorial 9**

1. Which of this keyword must be used to inherit a class?

a) super b) this c) extent d) extends

2. Which of these keywords is used to refer to member of base class from a sub class if they have the same name in both classes?

a) upper b) super c) this d) None of the mentioned

3. Which of these is correct way of inheriting class A by class B?

a) class B + class A {}

b) class B inherits class A {}

c) class B extends A {}

d) class B extends class A {}

4. What is the process of defining a method in subclass having same name & type signature as a method in its superclass?

a) Method overloading b) Method overriding c) Method hiding d) None of the mentioned

5. Which of these is supported by method overriding in Java?

a) Abstraction b) Encapsulation c) Polymorphism d) None of the mentioned

6. Inheritance represents what kind of relationship?

a) none of these b) is-a c) has-a d) was-a

**What is the output of the following program?**

1. **class** addition {

**int** a = 10;

**int** b = 5;

}

**class** Inheritance **extends** addition {

**int** c;

**void** add() {

c = a + b;

System.***out***.println("the sum =" + c);

}

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

Inheritance x = **new** Inheritance();

x.add();

}

}

1. **class** addition1 {

**void** init() {

**int** a = 10;

**int** b = 5;

}

}

**class** Demo1 **extends** addition1 {

**int** c;

**void** add() {

c = a + b;

System.***out***.println("the sum =" + c);

}

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

Demo1 x = **new** Demo1();

x.add();}}

1. **class** addition1 {

**void** init() {

**int** a = 10;

**int** b = 5;

}

}

**class** Demo1 **extends** addition1 {

**int** c;

**void** add() {

c = a + b;

System.***out***.println("the sum =" + c);

}

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

Demo1 x = **new** Demo1();

x.init();

x.add();

}

}

1. **class** Base {

**public** **void** show() {

System.***out***.println("Base::show() called");

}

}

**class** Derived **extends** Base {

**public** **void** show() {

System.***out***.println("Derived::show() called");

}

}

**public** **class** Main {

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

Derived b = **new** Derived();

b.show();

}

}

1. **class** A {

**int** i;

**void** display() {

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

}

}

**class** B **extends** A {

**int** j;

**void** display() {

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

}

}

**class** inheritance\_demo {

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

B obj = **new** B();

obj.i = 1;

obj.j = 2;

obj.display();

}

}

1. **class** Person {

**void** message() {

System.***out***.println("This is person class");

}

}

**class** Student **extends** Person {

**void** message() {

System.***out***.println("This is student class");

}

**void** display() {

message();

**super**.message();

}

}

**class** Test {

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

Student s = **new** Student();

// calling display() of //Student

s.display();

}

}

7)

**class** A {

String s = "Class A";

}

**class** B **extends** A {

String s = "Class B";

B() {

System.***out***.println(**super**.s);

}

}

**class** C **extends** B {

String s = "Class C";

C() {

System.***out***.println(**super**.s);

}

}

**public** **class** Abc

{

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

C c = **new** C();

System.***out***.println(c.s);

}

}

8)

**class** A {

{

System.***out***.println(1);

}

}

**class** B **extends** A {

{

System.***out***.println(2);

}

}

**class** C **extends** B {

{

System.***out***.println(3);

}

}

**public** **class** Abc {

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

C c = **new** C();

}

}

9)

**class** Grandparent {

**public** **void** Print() {

System.***out***.println("Grandparent's Print()");

}

}

**class** Parent **extends** Grandparent {

**public** **void** Print() {

**super**.Print();

System.***out***.println("Parent's Print()");

}

}

**class** Child **extends** Parent {

**public** **void** Print() {

**super**.Print();

System.***out***.println("Child's Print()");

}

}

**public** **class** abcde {

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

Child c = **new** Child();

c.Print();

}

}