Object Class have total 12 method

public        String       toString**()**

public        int          hashCode**()**

public        boolean      equals**(** Obj **)**

public        final        getclass**()**

public        final        void      notify**()**

public        final        void      notifyAll**()**

public        final        void      wait**()**

public        final        void      wait**(**long**)**

public        final        void      wait**(**long**,** int**)**

protected     void         finalize**()**

protected     Object       clone**()**

private       Static        registerNatives**().**

1. **How many classes we can’t override of object classes?**

Ans : Total 7 Classes we can’t override.

1. getclass()
2. notify()
3. notifyAll()
4. wait()
5. wait(long)
6. wait(long , int )
7. registerNatives
8. **How many classes are final in Object class?**

Ans : There are total 6 Class are final in Object class.

1. getclass()
2. notify()
3. notifyAll()
4. wait()
5. wait(long)
6. wait(long , int )
7. registerNatives
8. **How many classes are protected in Object class?**

Ans : There are total 3 Class are protected in Object class.

1. toString()
2. equals()
3. hashCode()
4. **How many classes we can override of object classes?**

Ans : Total 5 Classes we can override.

1. toString()
2. hashCode()
3. equals()
4. finalize()
5. clone().
6. **Why we Over ride toString Method?**

**Ans :**

1. toString method print meaning less data.
2. To print the meaning full data we override toString Method.
3. Because toString method print **class\_Name@HDN**

**When we print Object that time toString method is called.**

**(Zero parameter).**

class Employee

**{**

    int id**;**

    String name**;**

**}**

class Q01\_Object\_Class\_toString

**{**

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

**{**

        Employee e1 **=** **new** Employee**();**

        System**.**out**.**println**(**e1**);**

        System**.**out**.**println**(**e1**.**toString**()** **);**

**}**

**}**

Employee@15db9742

Employee@15db9742

class Employee

**{**

    int id**;**

    String name**;**

    public String toString**()**

**{**

**return** "Ram"**;**

**}**

**}**

class Q02\_Object\_Class\_toString

**{**

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

**{**

        Employee e1 **=** **new** Employee**();**

        System**.**out**.**println**(**e1**);**

        System**.**out**.**println**(**e1**.**toString**()** **);**

**}**

**}**

Ram

Ram

class Employee

**{**

    int id**;**

    String name**;**

    public String toString**()**

**{**

**return** "Ram"**;**

**}**

**}**

class Q03\_Object\_Class\_getClass

**{**

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

**{**

        Employee e1 **=** **new** Employee**();**

        System**.**out**.**println**(**e1**.**getClass**()** **);**

**}**

**}**

class Employee

class Employee

**{**

    int id**;**

    String name**;**

    public String toString**()**

**{**

**return** "Ram"**;**

**}**

**}**

class Q04\_Object\_Class\_getName

**{**

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

**{**

        Employee e1 **=** **new** Employee**();**

        System**.**out**.**println**(**e1**.**getClass**()** **);**

        System**.**out**.**println**(**e1**.**getClass**().**getName**()** **);**

**}**

**}**

class Employee

Employee

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public String toString**()**

**{**

**return** "Id : " **+** id **+** ", Name : " **+** name **;**

**}**

**}**

class Q05\_Object\_Class\_toString

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "A"**);**

        Employee e2 **=** **new** Employee**(**102**,** "B"**);**

        System**.**out**.**println**(**e1**);**

        System**.**out**.**println**(**e2**);**

        System**.**out**.**println**(**e1**.**toString**()** **);**

        System**.**out**.**println**(**e2**.**toString**()** **);**

**}**

**}**

Id : 101, Name : A

Id : 102, Name : B

Id : 101, Name : A

Id : 102, Name : B

class Employee

**{**

**}**

class Q06\_Object\_Class\_hashCode

**{**

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

**{**

        Employee e1 **=** **new** Employee**();**

        System**.**out**.**println**(**e1**);**

        System**.**out**.**println**(**e1**.**hashCode**()** **);**

        System**.**out**.**println**(** Integer**.**toHexString**(** e1**.**hashCode**()** **)** **);**

**}**

**}**

Employee@15db9742

366712642

15db9742

hashCode return  :  Decimal Number and

toString return  :  Class\_Name@Hexa\_Decimal\_Number

toHexString return :  Decimal to Hexa\_Decimal\_Number

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public String toString**()**

**{**

**return** "" **+** getClass**().**getName**()** **;**

**}**

**}**

class Q07\_Object\_Class\_toString

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "A"**);**

        System**.**out**.**println**(**e1**);**

**}**

**}**

Employee

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public String toString**()**

**{**

**return** "" **+** getClass**().**getName**()** **+** "@" **+** hashCode**();**

**}**

    public int hashCode**()**

**{**

**return**  id**;**

**}**

**}**

class Q08\_Object\_Class\_toString

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "A"**);**

        System**.**out**.**println**(**e1**);**

**}**

**}**

Employee@101

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public String toString**()**

**{**

**return** "" **+** getClass**().**getName**()** **+** "@" **+** Integer**.** toHexString**(** hashCode**()** **);**

**}**

    public int hashCode**()**

**{**

**return**  id**;**

**}**

**}**

class Q09\_Object\_Class\_toString

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "A"**);**

        System**.**out**.**println**(**e1**);**

**}**

**}**

Employee@65

Employee   @    65

class\_Name @ Hexa\_Decomal\_Number

class Q11\_Boolean\_Constructor

**{**

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

**{**

        Boolean b1 **=** **new** Boolean**(**"ram"**);**

        Boolean b2 **=** **new** Boolean**(**"Sita"**);**

        System**.**out**.**println**(**b1**);**

        System**.**out**.**println**(**b2**);**

**}**

**}**

false

false

Jab ham Boolean class ka Object create  karet tiem constructor me jo v pass karenege

true ke alawa baha ushe false hi lega

And Boolean class ne toString class ko esha tarah overRide kar ke rakha he ki jo pass karenege bahi return karenga

class Q12\_Boolean\_Constructor

**{**

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

**{**

        Boolean b1 **=** **new** Boolean**(**"ram"**);**

        Boolean b2 **=** **new** Boolean**(**"Sita"**);**

        System**.**out**.**println**(**b1**.**equals**(**b2**)** **);**

**}**

**}**

true

class Q13\_Boolean\_Constructor

**{**

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

**{**

        Boolean b1 **=** **new** Boolean**(**"true"**);**

        Boolean b2 **=** **new** Boolean**(**"True"**);**

        Boolean b3 **=** **new** Boolean**(**"TRUE"**);**

        System**.**out**.**println**(**b1**);**

        System**.**out**.**println**(**b2**);**

        System**.**out**.**println**(**b3**);**

**}**

**}**

true

true

true

Ham COnstructor me value kese v pass Kare ushe bo true hi lega.

\

1. Super class ka reference variable sub-class ke object ko store kar sakta he.

2. Compile time par hamesha reference variable ke corresponding check hota he,

   Runtime par Object ke corresponding working hogi.

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q14\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a1 **=** **new** A**();**         //Compile + Run

        A a2 **=** **new** B**();**         //Compile + Run

        A a3 **=** **new** C**();**         //Compile + Run

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Done

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q15\_Object\_Class\_Object\_Referance

**{**

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

**{**

        B b **=** **new** A**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

    B b = new A();

error: incompatible types: A cannot be converted to B

                B b = new A();

                      ^

if Case :

    C c = new A();

error: incompatible types: A cannot be converted to C

                C c = new A();

                      ^

if Case :

    B b = new C();

error: incompatible types: C cannot be converted to B

                B b = new C();

                      ^

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q16\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** B**();**

        B b **=** a**;**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

error: incompatible types: A cannot be converted to B

                B b = a;

                      ^

if Case :

    A a = new B();

    B b = (B)a;         //COmpile + Run     :   work

Done

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q17\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** B**();**

        B b **=** **(**B**)**a**;**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Exception in thread "main" java.lang.ClassCastException: C cannot be cast to B

IF  case - 1

A a = new A();

B b = (B)a;

Exception in thread "main" java.lang.ClassCastException: A cannot be cast to B

IF  case - 2

A a = new C();

B b = (B)a;

Exception in thread "main" java.lang.ClassCastException: A cannot be cast to B

IF  case - 3

A a = new B();

B b = (B)a;

Done

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q18\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

        B b **=** **new** B**();**

        b**.**show**();**

**}**

**}**

Class A

Class B

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q19\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** B**();**

        a**.**show**();**

**}**

**}**

Class B

class A

**{}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q20\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** B**();**

        a**.**show**();**

**}**

**}**

error: cannot find symbol

                a.show();

                 ^

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{}**

class Q21\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** B**();**

        a**.**show**();**

**}**

**}**

Class A

class A

**{}**

class B **extends** A

**{}**

class Q22\_Object\_Class\_Object\_Referance

**{**

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

**{**

        A a **=** **new** B**();**

        a**.**show**();**

**}**

**}**

error: cannot find symbol

                a.show();

                 ^

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class C **extends** B

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class C"**);**

**}**

**}**

class Q23\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 01

Class C

// 02

Class C

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class C **extends** B

**{}**

class Q24\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 01

Class B

// 02

Class B

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

**}**

class C **extends** B

**{**

**}**

class Q25\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 01

Class A

// 02

Class A

class A

**{**

**}**

class B **extends** A

**{**

**}**

class C **extends** B

**{**

**}**

class Q26\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 01

error: cannot find symbol

                b.show();

                 ^

  symbol:   method show()

// 02

error: cannot find symbol

                a.show();

                 ^

class A

**{**

**}**

class B **extends** A

**{**

**}**

class C **extends** B

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class C"**);**

**}**

**}**

class Q27\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 01

error: cannot find symbol

                b.show();

                 ^

  symbol:   method show()

// 02

error: cannot find symbol

                a.show();

                 ^

class A

**{**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class C **extends** B

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class C"**);**

**}**

**}**

class Q28\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 02

error: cannot find symbol

                a.show();

                 ^

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

**}**

class C **extends** B

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class C"**);**

**}**

**}**

class Q29\_Object\_Class\_Object\_Referance

**{**

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

**{**

        // 01

        B b **=** **new** C**();**

        b**.**show**();**

        // 02

        A a **=** **new** C**();**

        a**.**show**();**

**}**

**}**

// 01

Class C

// 02

Class C

instanceof keyword ka use parent - child relation me hi kar sakte he.

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q30\_Object\_Class\_instanceof

**{**

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

**{**

        A a **=** **new** A**();**

        B b **=** **new** B**();**

        C c **=** **new** C**();**

        // 01

        // System.out.println(a instanceof A);

        // System.out.println(b instanceof A);

        // System.out.println(c instanceof A);

        // 02

        // System.out.println (a instanceof B);

        // System.out.println(b instanceof B);

        // System.out.println(c instanceof B);

        // 03

        System**.**out**.**println**(**a **instanceof** C**);**

        // System.out.println(b instanceof C);

        System**.**out**.**println**(**c **instanceof** C**);**

**}**

**}**

// 01

true

true

true

// 02

false

true

error: incompatible types: C cannot be converted to B

    System.out.println(c instanceof B);

                     ^

// 03

false

error: incompatible types: B cannot be converted to C

    System.out.println(b instanceof C);

                       ^

true

**instanceof keyword ka use parent - child relation me hi kar sakte he.**

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q31\_Object\_Class\_instanceof

**{**

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

**{**

        Object o1 **=** **new** Object**();**

        String s1 **=** **new** String**();**

        StringBuffer b1 **=** **new** StringBuffer**();**

        // 01

        System**.**out**.**println**(**o1 **instanceof** Object**);**

        System**.**out**.**println**(**s1 **instanceof** Object**);**

        System**.**out**.**println**(**b1 **instanceof** Object**);**

        // 02

        System**.**out**.**println**(**o1 **instanceof** String**);**

        System**.**out**.**println**(**s1 **instanceof** String**);**

        // System.out.println(b1 instanceof String);

        // 03

        System**.**out**.**println**(**o1 **instanceof** StringBuffer**);**

        // System.out.println(s1 instanceof StringBuffer);

        System**.**out**.**println**(**b1 **instanceof** StringBuffer**);**

**}**

**}**

// 01

true

true

true

// 02

false

true

error: incompatible types: StringBuffer cannot be converted to String

    System.out.println(b1 instanceof String);

                       ^

// 03

false

 error: incompatible types: String cannot be converted to StringBuffer

        System.out.println(s1 instanceof StringBuffer);

                           ^

True

class A

**{}**

class B **extends** A

**{}**

class C **extends** A

**{}**

class Q32\_Object\_Class\_instanceof

**{**

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

**{**

        Object o1 **=** **new** Object**();**

        String s1 **=** **new** String**();**

        StringBuffer b1 **=** **new** StringBuffer**();**

        // 01

        System**.**out**.**println**(**o1 **instanceof** Object**);**

        System**.**out**.**println**(**s1 **instanceof** Object**);**

        System**.**out**.**println**(**b1 **instanceof** Object**);**

        // 02

        System**.**out**.**println**(**o1 **instanceof** Object**);**

        System**.**out**.**println**(**s1 **instanceof** Object**);**

        System**.**out**.**println**(**b1 **instanceof** Object**);**

        // 03

        System**.**out**.**println**(**o1 **instanceof** Object**);**

        System**.**out**.**println**(**s1 **instanceof** Object**);**

        System**.**out**.**println**(**b1 **instanceof** Object**);**

**}**

**}**

// 01

true

true

true

// 02

true

true

true

// 03

true

true

true

**null ke case me hamesha , false hi aayega.**

class Q33\_Object\_Class\_instanceof

**{**

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

**{**

        System**.**out**.**println**(null** **instanceof** Object**);**

        System**.**out**.**println**(null** **instanceof** Object**);**

        System**.**out**.**println**(null** **instanceof** Object**);**

**}**

**}**

false

false

false

class Q34\_Object\_Class\_equals

**{**

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

**{**

        Double a **=** 10.8**,** b **=** 10.8**;**

        Double x **=** 100.8**;**

        Double y **=** 100.8**;**

        System**.**out**.**println**(** a **==** b**);**

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

        System**.**out**.**println**(** a**.**equals**(**b**)** **);**

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

**}**

**}**

false

false

true

true

String class ki .equals() method content compare ka kam karti he.

( == ) operator referance ko compare karta he.

class Q35\_Object\_Class\_equals

**{**

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

**{**

        String s1 **=** **new** String**(**"Ram"**);**

        String s2 **=** **new** String**(**"Ram Ji"**);**

        String s3 **=** **new** String**(**"Ram"**);**

        System**.**out**.**println**(** s1 **==** s2 **);**

        System**.**out**.**println**(** s1 **==** s3 **);**

        System**.**out**.**println**(** s2 **==** s3 **);**

        System**.**out**.**println**(** s1**.**equals**(**s2**)** **);**

        System**.**out**.**println**(** s2**.**equals**(**s3**)** **);**

        System**.**out**.**println**(** s1**.**equals**(**s3**)** **);**

**}**

**}**

false

false

false

false

false

true

String class ki .equals() method content compare ka kam karti he.

( == ) operator referance ko compare karta he.

class Q36\_Object\_Class\_equals

**{**

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

**{**

        StringBuffer s1 **=** **new** StringBuffer**(**"Ram"**);**

        StringBuffer s2 **=** **new** StringBuffer**(**"Ram Ji"**);**

        StringBuffer s3 **=** **new** StringBuffer**(**"Ram"**);**

        System**.**out**.**println**(** s1 **==** s2 **);**

        System**.**out**.**println**(** s1 **==** s3 **);**

        System**.**out**.**println**(** s2 **==** s3 **);**

        System**.**out**.**println**(** s1**.**equals**(**s2**)** **);**

        System**.**out**.**println**(** s2**.**equals**(**s3**)** **);**

        System**.**out**.**println**(** s1**.**equals**(**s3**)** **);**

**}**

**}**

false

false

false

false

false

false

StringBuffer class ki .equals() method nai he that's by

Object class li equals or ( == ) operator referance ko compare karta he.

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public String toString**()**

**{**

**return** "Id : " **+** id **+** ", Name : " **+** name**;**

**}**

**}**

class Q37\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        System**.**out**.**println**(** e1 **);**

        System**.**out**.**println**(** e2 **);**

        System**.**out**.**println**(** e3 **);**

        System**.**out**.**println**(** e1 **==** e2 **);**

        System**.**out**.**println**(** e1 **==** e3 **);**

        System**.**out**.**println**(** e2 **==** e3 **);**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);**

        System**.**out**.**println**(** e2**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**

**}**

**}**

Id : 101, Name : Ram

Id : 102, Name : Sita

Id : 101, Name : Ram

false

false

false

false

false

false

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

**return** **true;**

**}**

**}**

class Q38\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        System**.**out**.**println**(** e1 **==** e2 **);**

        System**.**out**.**println**(** e1 **==** e3 **);**

        System**.**out**.**println**(** e2 **==** e3 **);**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);**

        System**.**out**.**println**(** e2**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**

**}**

**}**

false

false

false

true

true

true

Object class ki equals method referance comparion ka kam karti he.

ham chahte he content compare ho esha purpose se equals method ko overRide karte he.

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

        System**.**out**.**println**(**o**.**id**);**

**return** **true;**

**}**

**}**

class Q39\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);**

        System**.**out**.**println**(** e2**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e2**.**equals**(**e4**)** **);**

        System**.**out**.**println**(** e3**.**equals**(**e4**)** **);**

**}**

**}**

101

true

101

true

101

true

102

true

102

true

101

true

if Case :

    public boolean equals(Object o)

    {

        System.out.println(o.id);

        return true;

    }

    error: cannot find symbol

        System.out.println(o.id);

                           ^

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

        Employee e **=** **(**Employee**)**o**;**

**if(**id **==** e**.**id **&&** name**.**equals**(**e**.**name**)** **)**

**{**

**return** **true;**

**}**

**return** **false;**

**}**

**}**

class Q40\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);**

        System**.**out**.**println**(** e2**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e2**.**equals**(**e4**)** **);**

        System**.**out**.**println**(** e3**.**equals**(**e4**)** **);**

**}**

**}**

false

true

true

false

false

true

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

        Employee e **=** **(**Employee**)**o**;**

**if(**id **==** e**.**id **&&** name**.**equals**(**e**.**name**)** **)**

**{**

**return** **true;**

**}**

**return** **false;**

**}**

**}**

class Q41\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);**

        // System.out.println( e3.equals(101) );

        // System.out.println( e3.equals("Ram") );

        // System.out.println( e3.equals(null) );

**}**

**}**

false

true

true

java.lang.ClassCastException: java.lang.Integer can’t be cast to Employee

java.lang.ClassCastException: java.lang.String  can’t be cast to Employee

java.lang.NullPointerException

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

**}**

class Q42\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**101**)** **);**

        System**.**out**.**println**(** e1**.**equals**(**"Ram"**)** **);**

        System**.**out**.**println**(** e1**.**equals**(null)** **);**

**}**

**}**

false

false

true

false

false

false

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

**try**

**{**

            Employee e1 **=** **(**Employee**)**o**;**

**if(** id **==** e1**.**id **&&** name**.**equals**(**e1**.**name**)** **)**

**return** **true;**

**else**

**return** **false;**

**}**

**catch(**Exception e**)**

**{**

**return** **false;**

**}**

**}**

**}**

class Q43\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);** false

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);**  true

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);** true

        System**.**out**.**println**(** e3**.**equals**(**101**)** **);** false

        System**.**out**.**println**(** e3**.**equals**(**"Ram"**)** **);** false

        System**.**out**.**println**(** e3**.**equals**(null)** **);** false

**}**

**}**

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

**if(** o **instanceof** Employee**)**

**{**

            Employee e1 **=** **(**Employee**)**o**;**

**if(** id **==** e1**.**id **&&** name**.**equals**(**e1**.**name**)** **)**

**return** **true;**

**else**

**return** **false;**

**}**

**else**

**return** **false;**

**}**

**}**

class Q44\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);** false

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);** true

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);** true

        System**.**out**.**println**(** e3**.**equals**(**101**)** **);** false

        System**.**out**.**println**(** e3**.**equals**(**"Ram"**)** **);** false

        System**.**out**.**println**(** e3**.**equals**(null)** **);** false

**}**

**}**

class Employee

**{**

    int id**;**

    String name**;**

    Employee**(**int id**,** String name**)**

**{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**}**

    public boolean equals**(**Object o**)**

**{**

**if(this** **==** o **)**    **return** **true;**

**if(** o **instanceof** Employee**)**

**{**

            Employee e1 **=** **(**Employee**)**o**;**

**if(** id **==** e1**.**id **&&** name**.**equals**(**e1**.**name**)** **)**

**return** **true;**

**else**

**return** **false;**

**}**

**else**

**return** **false;**

**}**

**}**

class Q45\_Object\_Class\_equals

**{**

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

**{**

        Employee e1 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e2 **=** **new** Employee**(**102**,** "Sita"**);**

        Employee e3 **=** **new** Employee**(**101**,** "Ram"**);**

        Employee e4 **=** e1**;**

        System**.**out**.**println**(** e1**.**equals**(**e2**)** **);** false

        System**.**out**.**println**(** e1**.**equals**(**e3**)** **);** true

        System**.**out**.**println**(** e1**.**equals**(**e4**)** **);** true

        System**.**out**.**println**(** e3**.**equals**(**101**)** **);** false

        System**.**out**.**println**(** e3**.**equals**(**"Ram"**)** **);** false

        System**.**out**.**println**(** e3**.**equals**(null)** **);** false

**}**

**}**

class Q46\_Object\_Class\_Operator

**{**

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

**{**

        String s1 **=** **new** String**(**"Ram"**);**

        StringBuffer s2 **=** **new** StringBuffer**(**"Ram"**);**

        System**.**out**.**println**(**s1 **==** s2**);**

**}**

**}**

error: incomparable types: String and StringBuffer

                System.out.println(s1 == s2);

                                      ^

(==) Operator Parent child relation me hi work karte he.

class Q47\_Object\_Class\_Operator

**{**

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

**{**

        Object s1 **=** **new** String**(**"Ram"**);**

        Object s2 **=** **new** StringBuffer**(**"Ram"**);**

        System**.**out**.**println**(**s1 **==** s2**);**

**}**

**}**

false

(==) Operator Parent child relation me hi work karte he.

--------------------------------------------------------------

class Q48\_Object\_Class\_Operator

**{**

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

**{**

        Object s1 **=** **new** String**(**"Ram"**);**

        Object s2 **=** **null;**

        Object s3 **=** s1**;**

        System**.**out**.**println**(**s1 **==** s2**);**

        System**.**out**.**println**(**s1 **==** s3**);**

**}**

**}**

false

true

--------------------------------------------------------------

class Q49\_Object\_Class\_Operator

**{**

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

**{**

        Object s1 **=** **new** String**(**"Ram"**);**

        Object s2 **=** **null;**

        System**.**out**.**println**(**s1 **==** s2**);**

**}**

**}**

false

--------------------------------------------------------------

Object Class Ki equals method ki definition ye he

public boolean equals**(**Object o**)**

**{**

**return (this** **==** o **) ;**

**}**

**Object class l=ki equals method exception handle karne ke liye na hi try-catch block use kiya he na hi instanceof ka use kiya he.**

|  |  |  |
| --- | --- | --- |
| **No.** | **.equals method** | **== operator** |
|  | .equals() method ko over ride kar sakte he. | == Operator ko over ride nai kar sakte he. |
|  | .equals() method content ko compare karta he. | == operator reference ko compare karta he. |