Q1: class Emp{  
 int id=10;

String name="SCOTT";

double salary=1000.0;  
 }

class Example{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:Emp@hc

Q2: class Emp{  
 int id=10;

String name="SCOTT";

double salary=1000.0;

public int toString(){

return id;

}  
 }

class Example{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:CERROR

Q3:class Emp{

int id=10;

String name="SCOTT";

double salary=1000.0;

public String toString(Object r){

return name;

}

}

class Sample{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:Emp@hc

Q4: class Emp{

int id=10;

String name="SCOTT";

double salary=1000.0;

public String toString(){

return name;

}

}

class Sample{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:SCOTT

Q5: class Emp{

int id=10;

String name="SCOTT";

double salary=1000.0;

public String toString(){

return id+","+name+","+salary;

}

}

class Sample{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:10,SCOTT,1000.0

Q6: class Emp{

int id=10;

String name="SCOTT";

double salary=1000.0;

public static String toString(){

return id+","+name+","+salary;

}

}

class Sample{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:CERROR

Q7: class Emp{

int id=10;

String name="SCOTT";

double salary=1000.0;

public static String toString(Object r){

return id+”,”+name+”,”+salary;

}

}

class Sample{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:CERROR

Q8: class Point{

int x=10,y=20;

public String toString(){

return x+","+y;

}

}

class Sample{

public static void main(String[] args) {

Point p=new Point();

System.out.println(p);

}

}

:10,20

Q9: class Point{

int x=10,y=20;

public String toString(){

return x;

}

}

class Sample{

public static void main(String[] args) {

Point p=new Point();

System.out.println(p);

}

}

:10

Q10: class Emp{

int id=10;

String name="SCOTT";

double salary=1000.0;

public final String toString(){

return id+","+name+","+salary;

}

}

class Sample{

public static void main(String[] args) {

Emp e=new Emp();

System.out.println(e);

}

}

:10,SCOTT,1000.0

Q11: class Emp{

private int id;

private String name;

private double salary;

public Emp(int id,String name,double salary){

this.id=id;

this.name=name;

this.salary=salary;

}

public final String toString(){

return id+","+name+","+salary;

}

}

class Sample{

public static void main(String[] args) {

Emp e1=new Emp(10,"SCOTT",1000.0);

Emp e2=new Emp(20,"TIGER",2000.0);

System.out.println(e1);

System.out.println(e2);

}

}

:

10,SCOTT,1000.0

20,TIGER,2000.0

Q12: class Numeric{

private int value=10;

public String toString(){

return value+"";

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric();

System.out.println(n1);

}

}

:10

Q13: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(20);

System.out.println(n1);

System.out.println(n2);

}

}

: 10,20

Q14: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(20);

System.out.println(n1.hashCode());

System.out.println(n2.hashCode());

}

}

:918221580

2055281021

Q15: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

System.out.println(n1.hashCode());

System.out.println(n2.hashCode());

System.out.println(n1);

System.out.println(n2);

}

}

: 10

10

10

10

Q16: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

System.out.println(n1.hashCode());

System.out.println(n2.hashCode());

System.out.println(System.identityHashCode(n1));

System.out.println(System.identityHashCode(n1));

}

}

:class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

System.out.println(n1.hashCode());

System.out.println(n2.hashCode());

System.out.println(System.identityHashCode(n1));

System.out.println(System.identityHashCode(n1));

}

}

:class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

System.out.println(n1.hashCode());

System.out.println(n2.hashCode());

System.out.println(System.identityHashCode(n1));

System.out.println(System.identityHashCode(n1));

}

}

: 10

10

918221580

918221580

Q17: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1 == n2);

System.out.println(n1 == n3);

}

}

:false

false

Q18: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+"";

}

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

: false

false

Q19: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){  
 return this==r;  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

: false

false

Q20: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){  
 return this.value==r.value;  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q21: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){  
 return hashCode()==r.hashCode();  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q22: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){  
 return hashCode()==r.hashCode();  
 }

public int hashCode(){

return value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q23: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){

if(r instanceof Numeric)

{

Numeric n=r;  
 return this.value == n.value;

}else

return false;  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q24: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){

if(r instanceof Numeric)

{

Numeric n=(Numeric)r;  
 return this.value == n.value;

}else

return false;  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q25: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){

if(r instanceof Numeric)

return this.value == (Numeric)r.value;

else

return false;  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q26: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){

if(r instanceof Numeric)

return this.value == ((Numeric)r).value;

else

return false;  
 }

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q27: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){

return this.value == ((Numeric)r).value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

Numeric n3=new Numeric(20);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(n3));

}

}

Q28: class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public boolean equals(Object r){

return this.value == ((Numeric)r).value;

}

}

class Sample{

public static void main(String[] args) {

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(10);

System.out.println(n1.equals(n2));

System.out.println(n1.equals(10)); }

}

Q29: Given

class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

//place here equals method

}

Which is a recommended way of overridding equals() method

inside Numeric class

a. public boolean equals(Object r){

return this.value == ((Numeric)r).value;

}

b. public boolean equals(Object r){

return this == r;

}

c. public boolean equals(Object r){

return this.value == r.value;

}

d. public boolean equals(Object r){

if(r instanceof Numeric)

return this.value == ((Numeric)r).value;

else

return false

}

Q30: class Point{

int x=10,y=20;

//place here overriding equals() method

}

Which is recommended way of overriding equals() method inside

Point class

* public boolean equals(Object r){  
   return this==r;  
  }
* public boolean equals(Object r){  
   return this.x==r.x && this.y==r.y;  
  }
* public boolean equals(Object r){  
   return this.x==((Point)r).x;  
  }
* public boolean equals(Object r){  
   return this.x==((Point)r).x && this.y==((Point)r).y;  
  }
* public boolean equals(Object r){

if(r instanceof Point)  
 return this.x==((Point)r).x && this.y==((Point)r).y;

else

return false;  
}

Q31: class Emp{

int id;

String name;

double salary;

public Emp(int id,String name,double salary){

this.id=id;

this.name=name;

this.salary=salary;

}

public boolean equals(Object e){  
 return this.id==e.id && this.name==e.name && this.salary==e.salary;  
 }

}

class Sample{

public static void main(String[] args) {

Emp e1=new Emp(10,”SCOTT”,1000.0);

Emp e2=new Emp(10,”SCOTT”,1000.0);

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

}

}

Q32: class Emp{

int id;

String name;

double salary;

public Emp(int id,String name,double salary){

this.id=id;

this.name=name;

this.salary=salary;

}

public boolean equals(Object e){

if(e instanceof Emp)  
 return this.id==((Emp)e).id && this.name==((Emp)e).name && this.salary==((Emp)e).salary;

else return false;  
 }

}

class Sample{

public static void main(String[] args) {

Emp e1=new Emp(10,new String(”SCOTT”),1000.0);

Emp e2=new Emp(10,”SCOTT”,1000.0);

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

}

}

Q33: class Emp{

int id;

String name;

double salary;

public Emp(int id,String name,double salary){

this.id=id;

this.name=name;

this.salary=salary;

}

public boolean equals(Object e){

if(e instanceof Emp)  
 return this.id==((Emp)e).id && this.name.equals(((Emp)e).name) && this.salary==((Emp)e).salary;

else return false;  
 }

}

class Sample{

public static void main(String[] args) {

Emp e1=new Emp(10,new String(”SCOTT”),1000.0);

Emp e2=new Emp(10,”SCOTT”,1000.0);

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

}

}

Q34: Which is the recommended way of overriding equals() method inside Emp class

class Emp{

int id;

String name;

double salary;

//place equals() method

}

* public boolean equals(Object e){  
   return (this.id==e.id) &&(this.name.equals(e.name))

&& (this.salary==e.salary);  
 }

* public boolean equals(Object e){

if(e instanceof Emp)  
 return this.id==((Emp)e).id && this.name.equals(((Emp)e).name) && this.salary==((Emp)e).salary;

else

return false;  
 }

* public boolean equals(Object e){

return this==e;

}

* public boolean equals(Object e){

return this==(Emp)e;

}

Q35: class Set{

private Object[] e;

private int c;

public Set(){

e=new Object[12];

c=0;

}

public boolean add(Object item){

for(int i=0;i<c;i++){

if(e[i].equals(item) && e[i].hashCode()==item.hashCode())

return false;

}

if(c==e.length){

Object[] t=new Object[c+5];

for(int i=0;i<e.length;i++)

t[i]=e[i];

e=t;

}

e[c++]=item;

return true;

}

public String toString(){

String s="";

for(int i=0;i<c;i++)

s=s+e[i]+",";

return s;

}

}

class Sample{

public static void main(String[] args) {

Set s=new Set();

s.add(10);

s.add(20);

s.add(30);

s.add(40);

System.out.println(s);

}

}

Q36: class Set{

private Object[] e;

private int c;

public Set(){

e=new Object[12];

c=0;

}

public boolean add(Object item){

for(int i=0;i<c;i++){

if(e[i].equals(item) && e[i].hashCode()==item.hashCode())

return false;

}

if(c==e.length){

Object[] t=new Object[c+5];

for(int i=0;i<e.length;i++)

t[i]=e[i];

e=t;

}

e[c++]=item;

return true;

}

public String toString(){

String s="";

for(int i=0;i<c;i++)

s=s+e[i]+",";

return s;

}

}

class Sample{

public static void main(String[] args) {

Set s=new Set();

s.add(10);

s.add(20);

s.add(10);

s.add(30);

System.out.println(s);

}

}

Q37: class Set{

private Object[] e;

private int c;

public Set(){

e=new Object[12];

c=0;

}

public boolean add(Object item){

for(int i=0;i<c;i++){

if(e[i].equals(item) && e[i].hashCode()==item.hashCode())

return false;

}

if(c==e.length){

Object[] t=new Object[c+5];

for(int i=0;i<e.length;i++)

t[i]=e[i];

e=t;

}

e[c++]=item;

return true;

}

public String toString(){

String s="";

for(int i=0;i<c;i++)

s=s+e[i]+",";

return s;

}

}

class Sample{

public static void main(String[] args) {

Set s=new Set();

s.add(1);

s.add(2);

s.add(3);

s.add(4);

s.add(5);

s.add(6);

s.add(7);

s.add(8);

s.add(9);

s.add(1);

System.out.println(s);

}

}

Q38:class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+””;

}

}

class Set{

private Object[] e;

private int c;

public Set(){

e=new Object[12];

c=0;

}

public boolean add(Object item){

for(int i=0;i<c;i++){

if(e[i].equals(item) && e[i].hashCode()==item.hashCode())

return false;

}

if(c==e.length){

Object[] t=new Object[c+5];

for(int i=0;i<e.length;i++)

t[i]=e[i];

e=t;

}

e[c++]=item;

return true;

}

public String toString(){

String s="";

for(int i=0;i<c;i++)

s=s+e[i]+",";

return s;

}

}

class Sample{

public static void main(String[] args) {

Set s=new Set();

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(20);

Numeric n3=new Numeric(10);

Numeric n4=new Numeric(30);

s.add(n1);

s.add(n2);

s.add(n3);

s.add(n4);

System.out.println(s);

}

}

Q39:class Numeric{

private int value;

public Numeric(int value){

this.value=value;

}

public String toString(){

return value+””;

}

public int hashCode(){

return value;

}

public boolean equals(Object r){

if(r instanceof Numeric)

return this.value == ((Numeric)r).value;

else

return false;

}

}

class Set{

private Object[] e;

private int c;

public Set(){

e=new Object[12];

c=0;

}

public boolean add(Object item){

for(int i=0;i<c;i++){

if(e[i].equals(item) && e[i].hashCode()==item.hashCode())

return false;

}

if(c==e.length){

Object[] t=new Object[c+5];

for(int i=0;i<e.length;i++)

t[i]=e[i];

e=t;

}

e[c++]=item;

return true;

}

public String toString(){

String s="";

for(int i=0;i<c;i++)

s=s+e[i]+",";

return s;

}

}

class Sample{

public static void main(String[] args) {

Set s=new Set();

Numeric n1=new Numeric(10);

Numeric n2=new Numeric(20);

Numeric n3=new Numeric(10);

Numeric n4=new Numeric(30);

s.add(n1);

s.add(n2);

s.add(n3);

s.add(n4);

System.out.println(s);

}

}