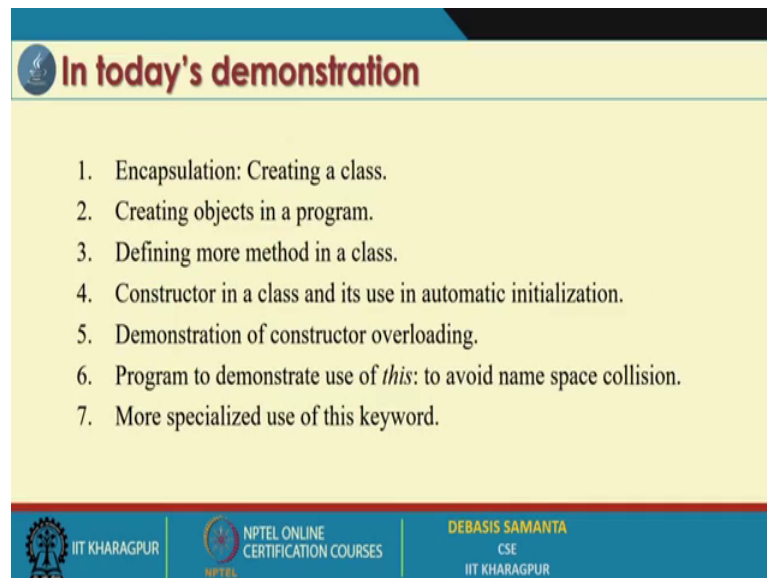


**Programming In Java**  
**Prof. Debasis Samanta**  
**Department of Computer Science Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture – 08**  
**Demonstration – III**

So, in the last module you have learned about encapsulation. Encapsulation in fact, is a very important concept is a very important object oriented paradigm. So, after learning basic concept of encapsulation so, today's we have planned a demo where we learn about different aspects in encapsulation.

(Refer Slide Time: 00:39)



**In today's demonstration**

1. Encapsulation: Creating a class.
2. Creating objects in a program.
3. Defining more method in a class.
4. Constructor in a class and its use in automatic initialization.
5. Demonstration of constructor overloading.
6. Program to demonstrate use of *this*: to avoid name space collision.
7. More specialized use of this keyword.

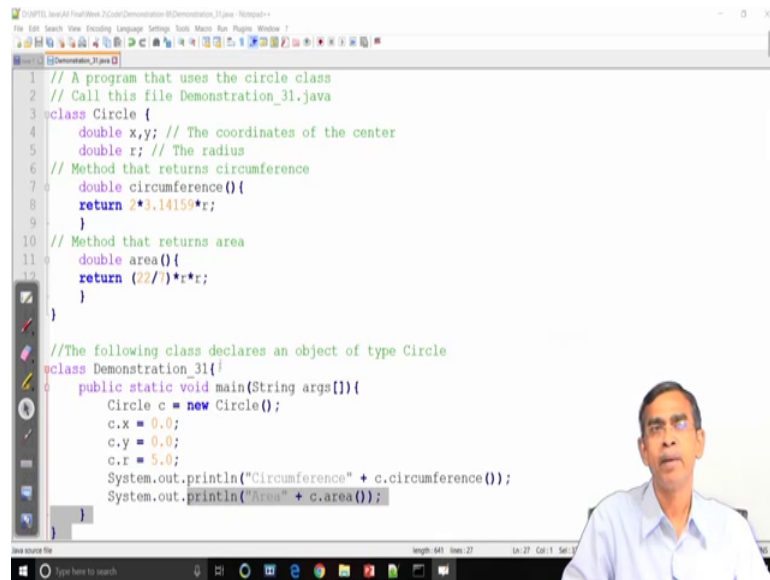
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA  
CSE  
IIT KHARAGPUR

So, our today's demonstration include how we can create a class and then for a class, how the different objects can be created and then how we can add many methods in a class so, that a class can take its completes form.

An important concept in encapsulation which is very essential for any program is that automatic object initialization. So, this concept is achieved in java by means of constructor. So, there is a concept of constructor will be demonstrated and also we will discuss about constructor overloading. So, it is a one part of the polymorphism in java concept so that many constructors can be plan. So, that an object can be initialized in a different ways and there are some concept that is very much used to resolve certain collision, it is called the name space collision resolution.

So, this is achieved by means of a special java keyword call the; this. So, we shall learn about this in this lecture. So, let us have the demo. So, first we will discuss about how we can create a class right. We will discussed a program very small program; let us have a very small program.

(Refer Slide Time: 02:05)



```
1 // A program that uses the circle class
2 // Call this file Demonstration_31.java
3 class Circle {
4     double x,y; // The coordinates of the center
5     double r; // The radius
6     // Method that returns circumference
7     double circumference(){
8         return 2*3.14159*r;
9     }
10    // Method that returns area
11    double area(){
12        return (22/7)*r*r;
13    }
14 }
15
16 //The following class declares an object of type Circle
17 class Demonstration_31{
18     public static void main(String args[]){
19         Circle c = new Circle();
20         c.x = 0.0;
21         c.y = 0.0;
22         c.r = 5.0;
23         System.out.println("Circumference" + c.circumference());
24         System.out.println("Area" + c.area());
25     }
26 }
```

In our theoretical discussion, we have discussed about one class called the circle. So, here we can see this is the; where is the highlighter. So, here we see we declare the class Circle here.

So, this class Circle has 3 data namely x, y and r all three data's are declared as double. In addition to this data also this class has two methods; one is circumference and another is area. So, the two methods are defined here. So, this completes the definition of a class called circle. Now, once this class is declared you can keep this class in a separate file or in the same file where the main class will include. Now here in the following we discuss the main class which is saved in the same file. The name of the main class in this case, is demonstration 31. Now in this class you see how we can create an object of the class type c circle.

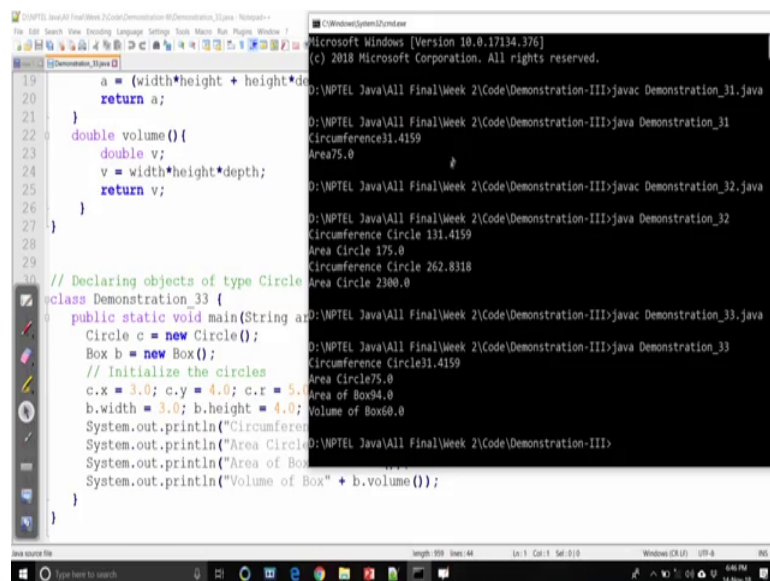
So, we create an object the name of the objective c and to create an object, you can note it the new operator which is used here. So, new and then again Circle and then within the parentheses, it is basically the general syntax to declare an object. So, with this statement we declare an object called the c the object is created; however, the different data that is

there for this object is yet to be initialized. So, in the following three statement we initialize the members x, y and r with the values 0.0, 0.0 and 5.0.

So, this completes the creation of an object, in this case the name of the objective is c. Now once this object is created, we can access any values in this object. So, in the next statement we just painting the circumference of the Circle that we have created. Now in order to access the method circumference, you note that we use c dot circumference. So, this means that we call the method circumference of the object c. In the next statement again, we call the method area of the object c. This means in this statement, we shall be able to print the area of the Circle which we have just now created

Now, let us have the demo. We will compile we will save this program as here the name of the main class is Demonstration underscore 31. We should save this file as the Demonstration underscore 31 dot java and then we can compile using java of c. So, let us see the compilation java c demonstration ok.

(Refer Slide Time: 05:27)



The screenshot displays a Java IDE with two windows. The left window shows the source code for a class named 'Demonstration\_31'. The code includes a 'volume()' method, a 'main()' method, and the creation of a 'Circle' object 'c' with radius 5.0 and a 'Box' object 'b' with dimensions 3.0 by 4.0. The right window shows the command prompt output, which includes the compilation command 'javac Demonstration\_31.java' and the execution command 'java Demonstration\_31'. The output shows the calculated circumference of the circle as 31.4159 and the area as 75.0.

```
19     a = (width*height + height*de
20     return a;
21 }
22 double volume(){
23     double v;
24     v = width*height*depth;
25     return v;
26 }
27 }
28 // Declaring objects of type Circle
29 class Demonstration_31 {
30     public static void main(String ar
31     Circle c = new Circle();
32     Box b = new Box();
33     // Initialize the circles
34     c.x = 3.0; c.y = 4.0; c.r = 5.0;
35     b.width = 3.0; b.height = 4.0;
36     System.out.println("Circumferen
37     System.out.println("Area Circle
38     System.out.println("Area of Box
39     System.out.println("Volume of Box" + b.volume());
40 }
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.376]
(c) 2018 Microsoft Corporation. All rights reserved.

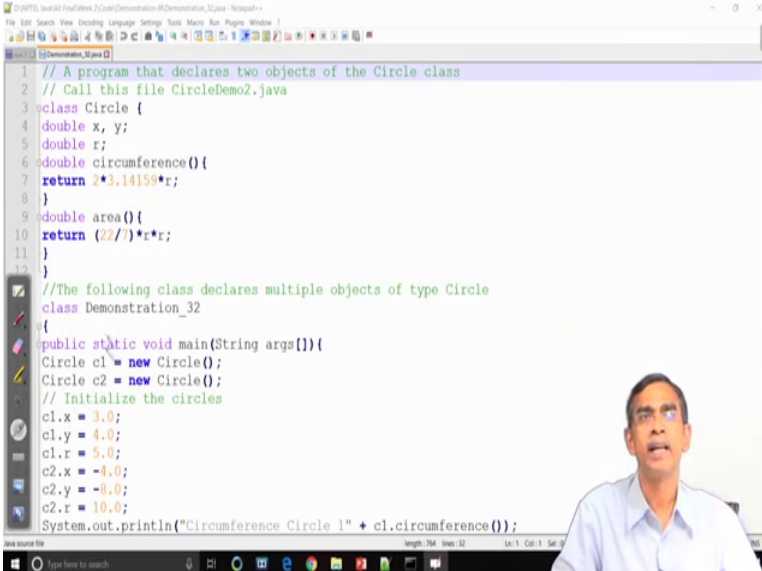
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_31.java
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_31
Circumference31.4159
Area75.0
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_32.java
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_32
Circumference Circle 131.4159
Area Circle 175.0
Circumference Circle 262.8318
Area Circle 2300.0
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_33.java
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_33
Circumference Circle31.4159
Area Circle75.0
Area of Box94.0
Volume of Box60.0
D:\MPTEL Java\All Final\Week 2\Code\Demonstration-III>
```

So, we just compile it as there is no error. So, compilation is successful, we can run this program using the java comment java is an interpreter. So, run this program and you see running this program and then circumference here in the Circle that is calculated as 39.4159 and the area is calculated 75.

Now, we have learned about how encapsulation is possible here in the form of encapsulation, we create a class Circle and for this class we created an object. Now here question is arised that whether we can create only one object in one program or many objects in a program. The answer is that multiple objects can be created for one class even multiple objects can be created for other classes also many classes also.

Now, in our next demonstration, we will see how for the same class say Circle the multiple circles can be created and let us have the demo here.

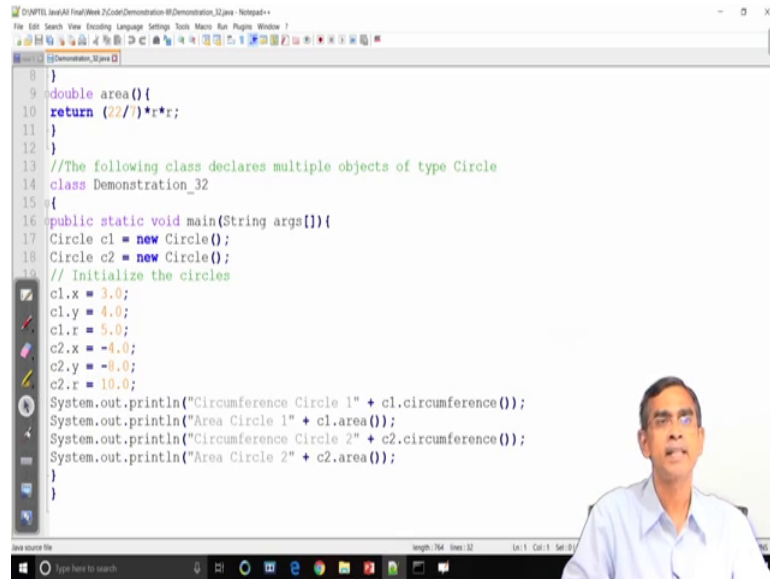
(Refer Slide Time: 06:41)



```
1 // A program that declares two objects of the Circle class
2 // Call this file CircleDemo2.java
3 class Circle {
4     double x, y;
5     double r;
6     double circumference(){
7         return 2*3.14159*r;
8     }
9     double area(){
10        return (22/7)*r*r;
11    }
12 }
13
14 //The following class declares multiple objects of type Circle
15 class Demonstration_32
16 {
17     public static void main(String args[]){
18         Circle c1 = new Circle();
19         Circle c2 = new Circle();
20         // Initialize the circles
21         c1.x = 3.0;
22         c1.y = 4.0;
23         c1.r = 5.0;
24         c2.x = -4.0;
25         c2.y = -8.0;
26         c2.r = 10.0;
27         System.out.println("Circumference Circle 1" + c1.circumference());
28     }
29 }
```

Again we use the same declaration of the Circle class here. It is as usual in the earlier one. Now only we change the main program main class. So, name this main class as Demonstration underscore 32. Now here we can see, we have created two objects c 1 and c 2. In the earlier example we have created only one object, here we have created two objects and for these two objects c 1 and c 2. We initialize its data values namely for the first object x y and r are initializes three 3.0, 4.0, 5.0.

(Refer Slide Time: 07:19)



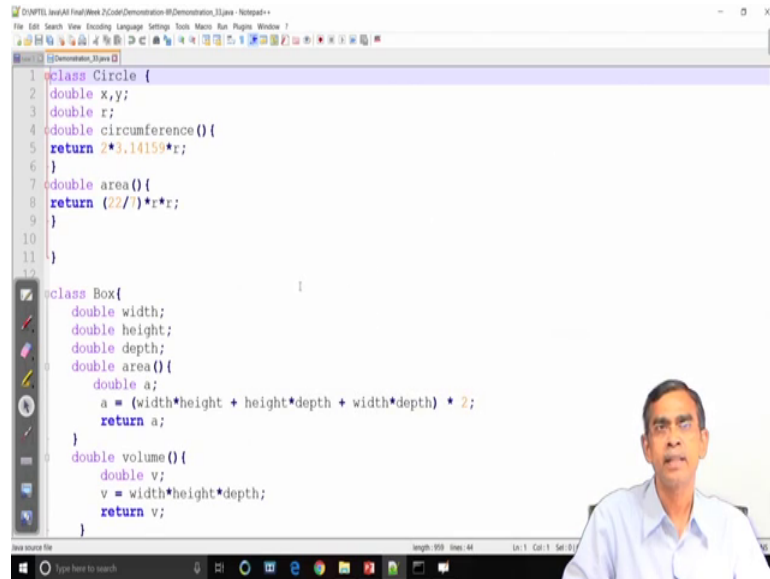
```
8 }
9 double area(){
10 return ((2*3.14)*r*r);
11 }
12 }
13 //The following class declares multiple objects of type Circle
14 class Demonstration_32
15 {
16 public static void main(String args[]){
17 Circle c1 = new Circle();
18 Circle c2 = new Circle();
19 // Initialize the circles
20 c1.x = 3.0;
21 c1.y = 4.0;
22 c1.r = 5.0;
23 c2.x = -4.0;
24 c2.y = -8.0;
25 c2.r = 10.0;
26 System.out.println("Circumference Circle 1" + c1.circumference());
27 System.out.println("Area Circle 1" + c1.area());
28 System.out.println("Circumference Circle 2" + c2.circumference());
29 System.out.println("Area Circle 2" + c2.area());
30 }
31 }
```

Whereas, for the second object c 2, we initially the same set of values with minus 4.0, minus 8.0, 10.0. Thus two objects are created and their data are also initialized.

Now, in the next 4 statements we see how we can access the different methods in these two objects namely circumference and its area. So, it is the same as earlier program accept that for two different objects we have used it. Now let us have the demo for this. So, in this case two objects are created and you will be able to get the values of the two objects here. So, same process we can compile it and then we can run it. So, now, we can see the two different objects and as the objects are the different their area and then circumference as printed area also different.

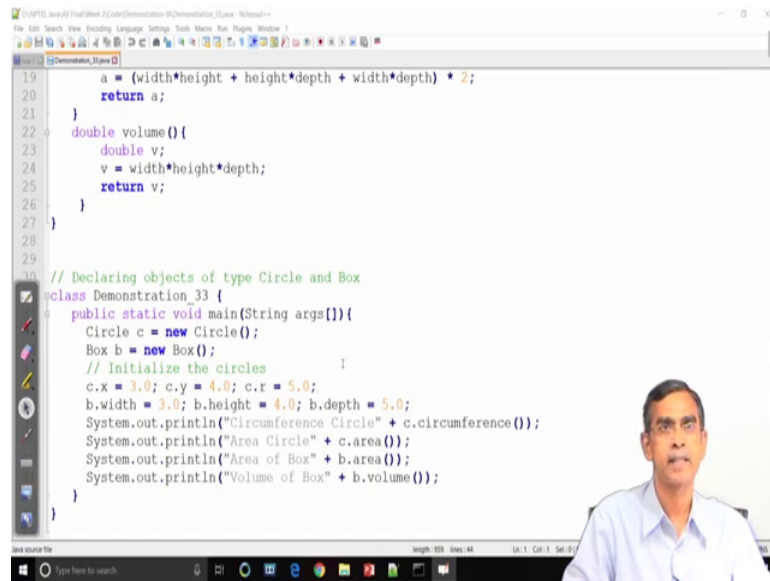
So, now we have understood that how for a class many objects can be created. In this case, we have considered to their maybe a large number of objects of the same class can be created. Now, in the our next demonstration is that whether one program can include more than one classes or not.

(Refer Slide Time: 08:37)

A screenshot of a Java IDE window titled "D:\NPTEL Java\Week 2\Code\Demonstration\2\Demonstration\_31.java - Notepad++". The code defines two classes: Circle and Box. The Circle class has attributes x, y, and r, and methods circumference() and area(). The Box class has attributes width, height, and depth, and methods area() and volume().

```
1 class Circle {
2     double x,y;
3     double r;
4     double circumference(){
5         return 2*3.14159*r;
6     }
7     double area(){
8         return (22/7)*r*r;
9     }
10 }
11 }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
26
```

(Refer Slide Time: 09:29)



```
19     a = (width*height + height*depth + width*depth) * 2;
20     return a;
21 }
22 double volume(){
23     double v;
24     v = width*height*depth;
25     return v;
26 }
27 }
28
29
30 // Declaring objects of type Circle and Box
31 class Demonstration_33 {
32     public static void main(String args[]){
33         Circle c = new Circle();
34         Box b = new Box();
35         // Initialize the circles
36         c.x = 3.0; c.y = 4.0; c.r = 5.0;
37         b.width = 3.0; b.height = 4.0; b.depth = 5.0;
38         System.out.println("Circumference Circle" + c.circumference());
39         System.out.println("Area Circle" + c.area());
40         System.out.println("Area of Box" + b.area());
41         System.out.println("Volume of Box" + b.volume());
42     }
43 }
```

On the other hand circumference is a unique method in the class Circle where is the volume is also unique method in the class Box. So, this way we create two classes namely Circle and box, we can save all these two classes in same file as the main class. Now let us have the look of the main class, we give the name of the main class as Demonstration underscore 33. This is the main class which includes the main method and now you see that in this main method, we create two objects one object of the type class Circle another object of the type class box. So, the two object c small c and small b are created here.

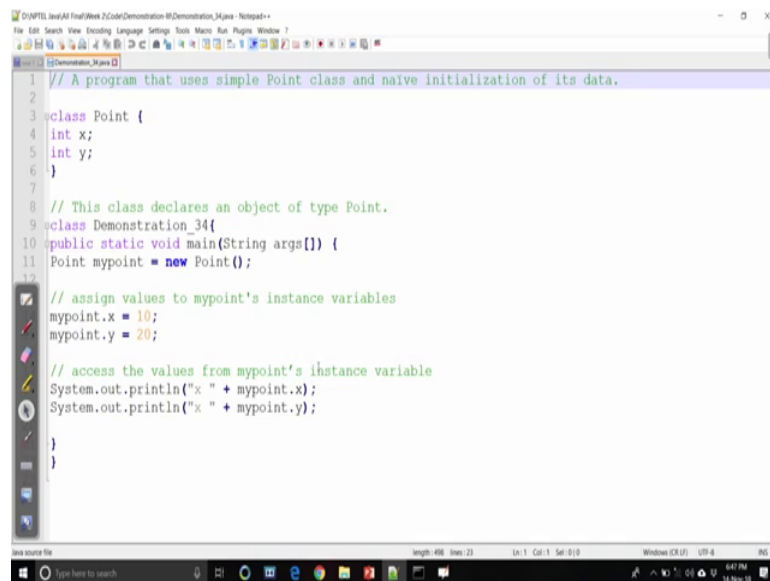
Now, again as it is as it was in the case of circle, we initialize the value. Here also the objects that we have created to initialize their values. So, here in the first statement, we create a we initialize the value of the Circle objects with 3.0, 4.0 and 5.0 whereas, for the Box object we initialize the value of width height and depth as the values 3.0, 4.0, 5.0. So, this completes creation of two different type of objects of two different classes.

Now, in the next four statements, we can access the different methods. As we have seen here for the Circle c, we access the area and circumference and for the Box b we access the area and volume. Now let us have the execution of this program, we will be able to see that the two objects they are defined in terms of their classes and then they can access by the main method. So, now, we can see that for the first two printout is basically the

display basically Circle circumference and area and next two output is for the Box area of the Box and volume of the Box

Now, we have understood about that how many classes can be created. It is not necessary that all the classes should be maintain in the same file in the main class. We will discussed about organization of the classes in your in our program that will be discussed in due time not now. So, now, let us see here we can see when we create an objects. So, it is our responsibility to initialize. The objects initialization of an object means initialization of all the member elements that belongs to the object. For example, for the Circle x y r for the Box width height and depth, we have to the initialization, but this initialization as we have done. In the last few examples little bit t d s because we have to do forcefully anyway, but the initialization can be done more sophisticated way in our next program. We will see how the initialization of the objects can be done in an easy way.

(Refer Slide Time: 12:49)



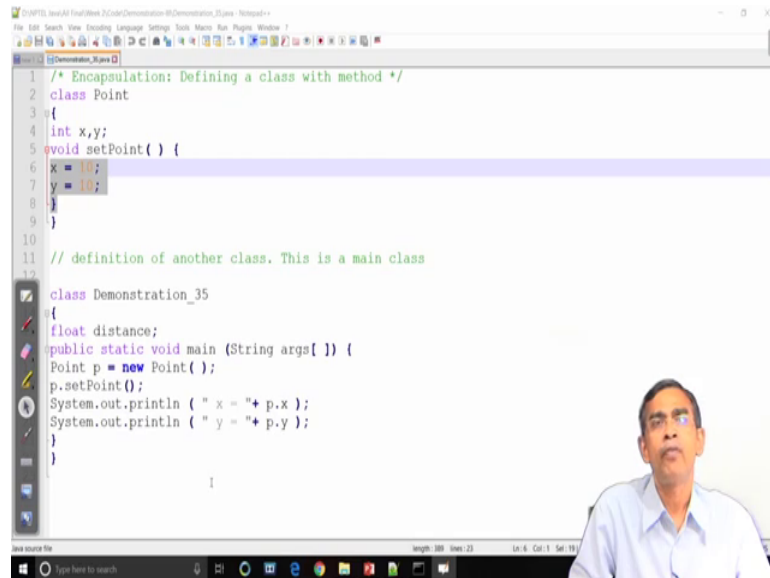
```
1 // A program that uses simple Point class and naive initialization of its data.
2
3 class Point {
4     int x;
5     int y;
6 }
7
8 // This class declares an object of type Point.
9 class Demonstration_34 {
10     public static void main(String args[]) {
11         Point mypoint = new Point();
12
13         // assign values to mypoint's instance variables
14         mypoint.x = 10;
15         mypoint.y = 20;
16
17         // access the values from mypoint's instance variable
18         System.out.println("x " + mypoint.x);
19         System.out.println("y " + mypoint.y);
20     }
21 }
```

So, here is a one example a 3.4 right let us see. So, now, we have declared here one, another class Point it has two member elements x and y declared as integer. It does not have any method in this case. Anyway we want to discuss about how this x y value can be initialized. Now here if you see, this is the usual procedure that we can create an object of type class and for this thing we can initialize the value from the main program. This is a usual procedure that you have learn earlier apart from this process.



Now, we will discuss about we can initialize by calling some methods. Now so, do to do this thing we have to declare a method in the class Point itself. Now let us see the next class 3.5 as a demo, we can see the class Point is little bit redefined with another method.

(Refer Slide Time: 13:41)



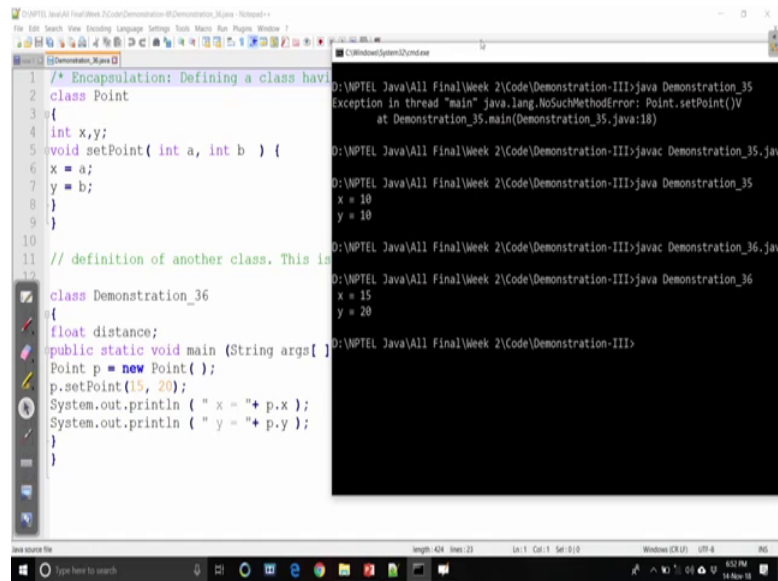
```
1 /* Encapsulation: Defining a class with method */
2 class Point
3 {
4     int x,y;
5     void setPoint() {
6         x = 10;
7         y = 10;
8     }
9 }
10
11 // definition of another class. This is a main class
12
13 class Demonstration_35
14 {
15     float distance;
16     public static void main (String args[] ) {
17         Point p = new Point() ;
18         p.setPoint();
19         System.out.println ( " x = "+ p.x );
20         System.out.println ( " y = "+ p.y );
21     }
22 }
```

The name of the method which we have include here set point. The setPoint has been planned to initialize its value. So, here example we can see the setPoint method has the 2 initialization statement x equals to 10 and y equals to 10.

Now, come to the you creation of an object. Now we create this object by main class here. The name of the main class is Demonstration underscore 35 and we have declared one variable distance ok. It will be it not useful here in this context anyway, but here see we create an object p of class Point is the next statement here and then we call the setPoint method is basically initialize the values for the object p and as you know by means of initialization one method setPoint, it will initialize as x equals 10 and y equals to 10.

Now, in the next two statement if we can print the value of the x and y of the object p, it will be printed like this. Now let us run this program.

(Refer Slide Time: 14:59)



```
1 /* Encapsulation: Defining a class having
2 class Point
3 {
4     int x,y;
5     void setPoint( int a, int b ) {
6         x = a;
7         y = b;
8     }
9 }
10
11 // definition of another class. This is
12
13 class Demonstration_36
14 {
15     float distance;
16     public static void main (String args[] )
17     {
18         Point p = new Point( );
19         p.setPoint(15, 20);
20         System.out.println ( " x = "+ p.x );
21         System.out.println ( " y = "+ p.y );
22     }
23 }
```

```
D:\WPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_35
Exception in thread "main" java.lang.NoSuchMethodError: Point.setPoint(J)V
    at Demonstration_35.main(Demonstration_35.java:18)

D:\WPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_35.java

D:\WPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_35
x = 10
y = 10

D:\WPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_36.java

D:\WPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_36
x = 15
y = 20

D:\WPTEL Java\All Final\Week 2\Code\Demonstration-III>
```

We are now executing this program. The program that Demonstration underscore 35 which basically includes the Point class declaration and then the setPoint the method their and then we will create an object of the type dot java. So, the object the program is now successfully compiled, we just going to run it 31.5 ok; fine.

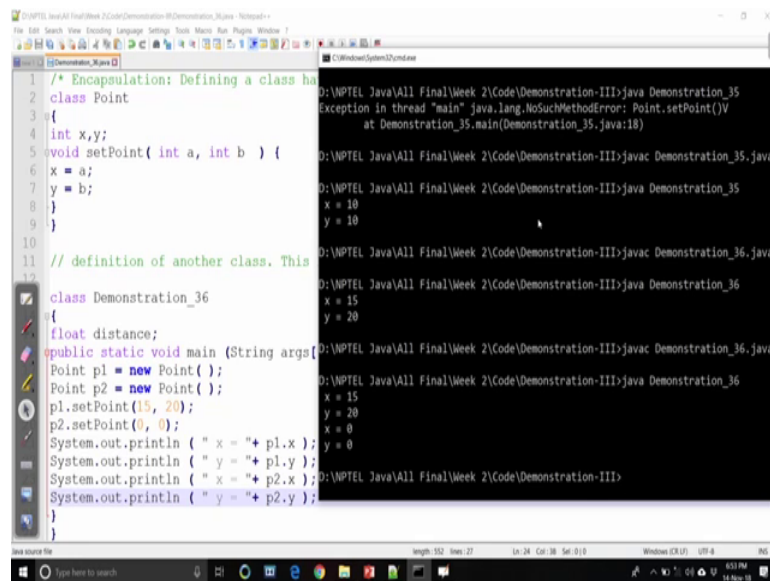
So, let us see here. So, here we can see the object is created and we did not do any initialization in the main method. We did the initialization via the setPoint method and the initialization initiation here x equals to 10 and y equals to 10. Now here we can see that initialization this method, but how at this method always initialize only with certain initialization. Here for example, x equals to 10 y equals to 10. This means that if we create any other objects, it will and we call the setPoint method for initially, we always initialize with the same 10. So, this is not a good and not desirable.

Now we will have a good method that how the installation object can be done with the different data values. So, in that case we have to again rewrite this function this method using passing argument. Now here is an example, we just slightly change this class declaration and setPoint method also passing two values a and b here a and b are the two values to be passed to this method. So, that a can be used to initialize the value of x the and b can be used to initialize the value of y and the rest of the program is same as earlier one, but only the difference you can see when we call this setPoint we pass 1520; that

means, in this case the objective p which we have created it can be initialized with its x value as 15 and y value as 20.

Now, let us run this program and then have a quick demo. So, 3.6, right. Now let us run this and we will see the object data member will be printed as 1520. Now let us come to the program, we can call this method or we can create another object say p 1 and p 2. We are creating two objects p 1 and p 2 here right; the p 1 and p 2 ok.

(Refer Slide Time: 17:47)



```
1 /* Encapsulation: Defining a class ha
2 class Point
3 {
4 int x,y;
5 void setPoint( int a, int b ) {
6 x = a;
7 y = b;
8 }
9 }
10
11 // definition of another class. This
12
13 class Demonstration_36
14 {
15 float distance;
16
17 public static void main (String args[]
18 {
19 Point p1 = new Point ();
20 Point p2 = new Point ();
21 p1.setPoint(15, 20);
22 p2.setPoint(0, 0);
23 System.out.println ( " x = "+ p1.x );
24 System.out.println ( " y = "+ p1.y );
25 System.out.println ( " x = "+ p2.x );
26 System.out.println ( " y = "+ p2.y );
27 }
28 }
29 }
```

```
D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_35
Exception in thread "main" java.lang.NoSuchMethodError: Point.setPoint(JV
at Demonstration_35.main(Demonstration_35.java:18)

D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_35.java

D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_35
x = 10
y = 10

D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_36.java

D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_36
x = 15
y = 20

D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration_36.java

D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration_36
x = 15
y = 20
x = 0
y = 0

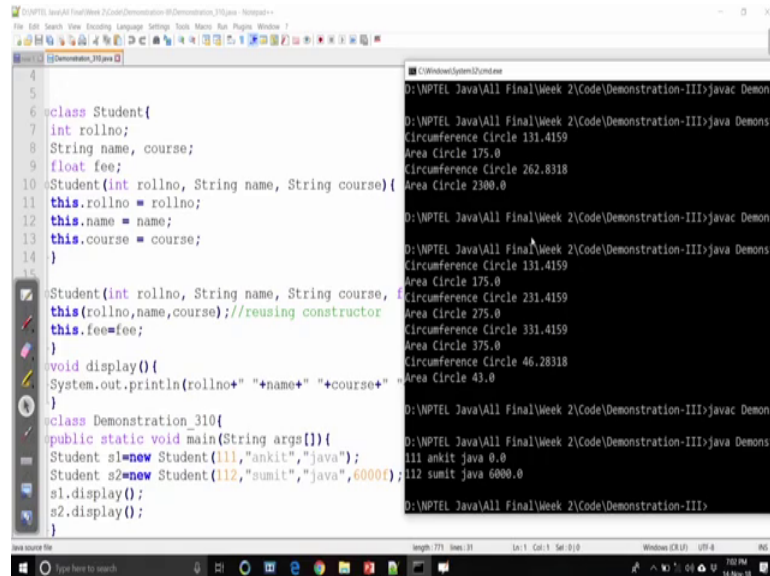
D:\VNPTEL Java\All Final\Week 2\Code\Demonstration-III>
```

Again setPoint we just on keep it. One another p 2, another say 0 and 0 p 1 p 2 0 p 1 and p 2 and 0 0 ok.

So, we create two objects one is 15, 20 as the value of x and y ' another is 0,0. Now again we can print it just two statements. So, we can we can print the values of two objects here right for first one is p 1 dot x p 1 dot y p 2 dot x p 2 dot y right. Now let us save this program. Now here we have created two objects and the two objects are initialized with their own values and then we use this main method to print their values by which is at. Now we can see that the two objects are created and they have in been initialized by their own values by means of setPoint method

Now, here is the one idea about we have to explicitly initialize the object by means of declaring and defining the method.

(Refer Slide Time: 18:53)

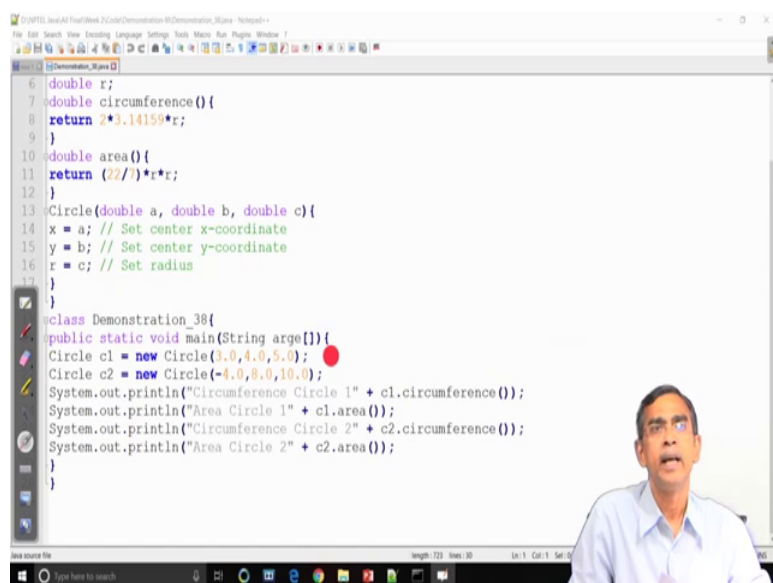


```
4
5
6 class Student{
7     int rollno;
8     String name, course;
9     float fee;
10    Student(int rollno, String name, String course){
11        this.rollno = rollno;
12        this.name = name;
13        this.course = course;
14    }
15
16    Student(int rollno, String name, String course,
17            this.rollno,name,course);//reusing constructor
18        this.fee=fee;
19    }
20    void display(){
21        System.out.println(rollno+" "+name+" "+course+"
22        ")
23    }
24    class Demonstration_310{
25    public static void main(String args[]){
26        Student s1=new Student(111,"ankit","java");
27        Student s2=new Student(112,"sumit","java",6000);
28        s1.display();
29        s2.display();
30    }
31    }
```

In java a unique features are available which basically help java programmer to automatically initialize the objects and that initialnce through at the time when the object is created itself. Now this concept the automatic initialization at the time of creating an object is called constructor. In our next example, we will see how the automatic object initialization using the concept of constructor is possible.

Now, let us have the program here not this program; go to the program number 3.8.

(Refer Slide Time: 19:39)



```
6 double r;
7 double circumference(){
8     return 2*3.14159*r;
9 }
10 double area(){
11     return (22/7)*r*r;
12 }
13 Circle(double a, double b, double c){
14     x = a; // Set center x-coordinate
15     y = b; // Set center y-coordinate
16     r = c; // Set radius
17 }
18 }
19
20 class Demonstration_38{
21 public static void main(String args[]){
22     Circle c1 = new Circle(3.0,4.0,5.0);
23     Circle c2 = new Circle(-4.0,8.0,10.0);
24     System.out.println("Circumference Circle 1" + c1.circumference());
25     System.out.println("Area Circle 1" + c1.area());
26     System.out.println("Circumference Circle 2" + c2.circumference());
27     System.out.println("Area Circle 2" + c2.area());
28 }
29 }
```

So, let us have a simple program which basically helps to automatically create the objects and then initialize it. Now let us have the program here. Now the class again we can continue our this demonstration with the class Circle this class Circle has the 3 member elements x y r two methods circumference and area.

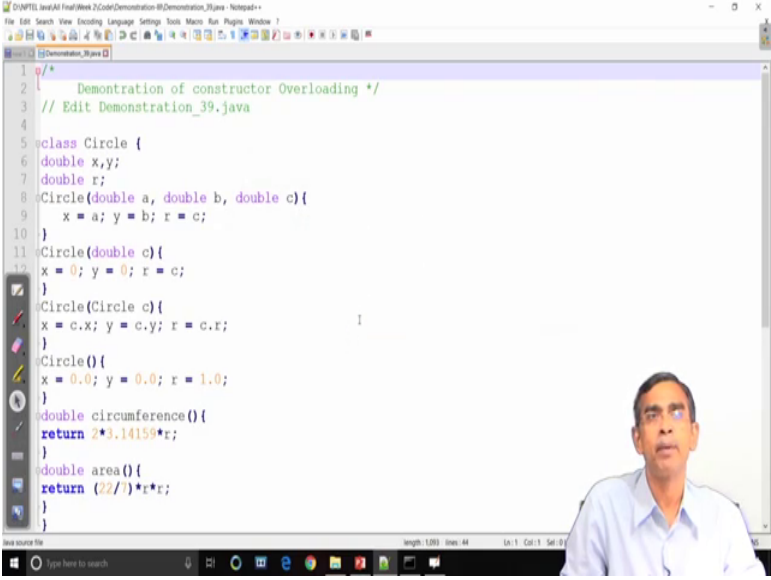
Now, we declare one method. This is a new method that we have added here in this class the name of this method. You see the is the same as the name of the class. This is a unique thing here the name of the method is same as the name of the class and also you can note in this case this method does not have any return type. So, this is called a constructor which is basically has the same name as the name of the class itself and it should not have any return type

Now, here the name of the constructor is there for Circle and this constructor is defined with 3 arguments. All these three arguments are of type double a b and c here. Basically a is the argument which will be passed to the value x, b is another argument whose value will be passed to y and c is the another argument whose value will be passed to r. This means that if we can create an object passing 3 values and they this 3 values will be used to initialize its member element. Now let us have the main class here after defining the Circle we can initialize the objects now here we see at the time of creating the object.

So, we are creating the object using the new operator here. Now in the first creation we create the object c 1 and we passed 3 values 3.0, 4.0, 5.0. This many means that constructor, it will call the method Circle which is the constructor method and pass this value to the Circle objects c 1. Similarly c 2 object will be initialize. After the initiation is over, the method will be accessed and then method will print the different values the according to the calculation or definition of the method.

Now, let us have the execution is the same as earlier. There is no different so far, the compilation execution is concerned; however, from the result point of view it is more sophisticated way of initialization. So, we have created the object at that time we have initialize the object at the time of creating the objects and then this is by means of constructor this concept is call constructor as I told you.

(Refer Slide Time: 22:39)



```
1  /*
2     Demonstration of constructor Overloading */
3     // Edit Demonstration_39.java
4
5     class Circle {
6     double x,y;
7     double r;
8     Circle(double a, double b, double c){
9     x = a; y = b; r = c;
10    }
11    Circle(double c){
12    x = 0; y = 0; r = c;
13    }
14    Circle(Circle c){
15    x = c.x; y = c.y; r = c.r;
16    }
17    Circle(){
18    x = 0.0; y = 0.0; r = 1.0;
19    }
20    double circumference(){
21    return 2*3.14159*r;
22    }
23    double area(){
24    return (2/7)*r*r;
25    }
26    }
```

Now, constructor has its more dimension. In the sense that it also gives a lot of flexibility to a programmer to initialize an object in a different way. Now if we if you want to initialize the object in a different way this then you have to define. So, many constructor for your own requirement Now this concept is called the constructor overloading, now here again let us come to the discussion of the class Circle in the last example we have discussed about how the automatic initialization by means of a constructor with three arguments we have defined it.

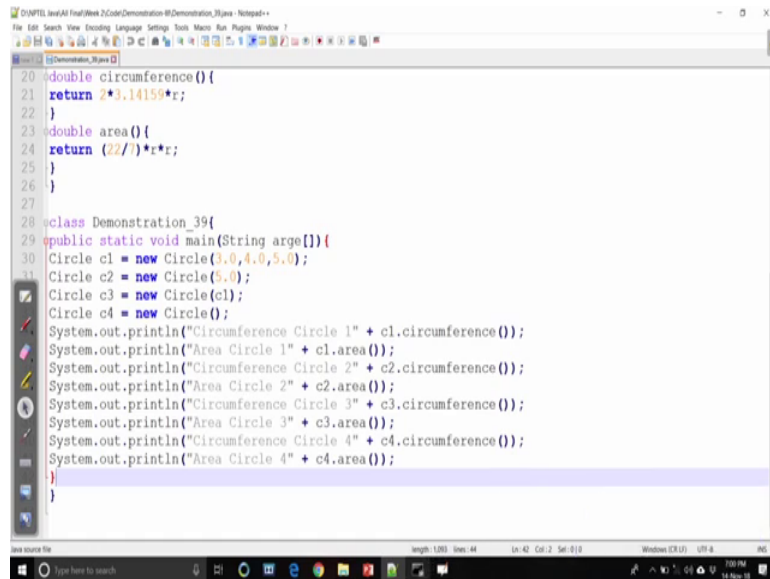
Now, in addition to this, we want to define another way of initializing. Here in the next constructor that we have declared. Here if you see, this constructor has only one argument double c and the constructor is defined like this if we call a constructor with one argument, then it will initialize their member elements with x equals to 0 y equals to 0 and r equals to 0 as for this definition. That mean if we create an object passing only 1 argument, then the java runtime compiler will automatically understand that this constructor should be invoke.

If we create an object passing 3 arguments, then the first constructor will be called. Now here again you see the third constructor that we have defined here. It is a little bit different; here we can pass and argument which is the object itself that is quite possible. So, here the Circle c another object which has been passed. if we passed it this means that the new object that will be created having the same data member as the Circle c

here. So, this is the one constructor; that means, a duplicate one object can be created duplicate object in the sense that they are different objects; however, same data values.

Another way is called the default constructor. If you define a constructor without passing and argument,t then it is called the default constructor.

(Refer Slide Time: 24:43)



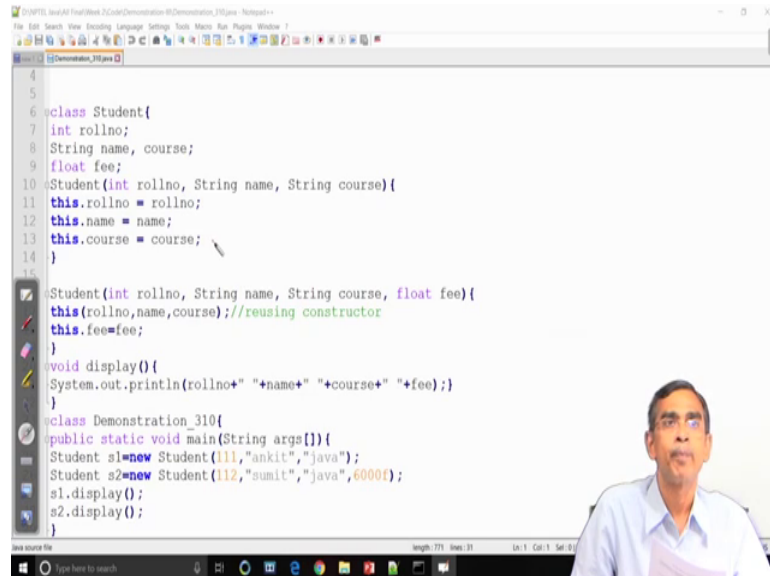
```
20 double circumference(){
21     return 2*3.14159*r;
22 }
23 double area(){
24     return (22/7)*r*r;
25 }
26 }
27
28 class Demonstration_39{
29     public static void main(String args[]){
30         Circle c1 = new Circle(5.0,4.0,5.0);
31         Circle c2 = new Circle(5.0);
32         Circle c3 = new Circle(c1);
33         Circle c4 = new Circle();
34
35         System.out.println("Circumference Circle 1" + c1.circumference());
36         System.out.println("Area Circle 1" + c1.area());
37         System.out.println("Circumference Circle 2" + c2.circumference());
38         System.out.println("Area Circle 2" + c2.area());
39         System.out.println("Circumference Circle 3" + c3.circumference());
40         System.out.println("Area Circle 3" + c3.area());
41         System.out.println("Circumference Circle 4" + c4.circumference());
42         System.out.println("Area Circle 4" + c4.area());
43     }
44 }
```

So, in this case the last constructor the fourth constructor in the series is called the default constructor. Here the default constructor means that x equals to 0.0, y equals to 0.0, r equals to 0.0. Now let us have the main program. So, main program is defined here by means of main class demonstration underscore 39. Now here we see we have created 4 objects and each objects are created to with their own constructor.

So, the first objects c one created with the first constructor second object with the second constructor and third object with the third constructor and finally, this one here ok. So, so, here if you see the c3 and c1 when we create the object c3 we pass the c1 c1 which is a which is an object of the class Circle as for the third constructor declaration. And the rest of the things are basically printing the different parameters area and circumference for the four Circle that we have created and definitely c1 and c1 c3 should give the same result. Now let us have the Demonstration. So, let us have the quick demo, here we are overloading constructor now. So, all 4 objects are created automatically and the objects are created at the time of creation of at the time of creating object; I mean declaration and by passing the values

So, this is about the overloading constructors. Now there are few more important and interesting facts about this class and then object constructor.

(Refer Slide Time: 26:31)



```
4
5
6 class Student{
7     int rollno;
8     String name, course;
9     float fee;
10    Student(int rollno, String name, String course){
11        this.rollno = rollno;
12        this.name = name;
13        this.course = course;
14    }
15
16    Student(int rollno, String name, String course, float fee){
17        this(rollno,name,course);//reusing constructor
18        this.fee=fee;
19    }
20
21    void display(){
22        System.out.println(rollno+" "+name+" "+course+" "+fee);
23    }
24
25    class Demonstration_310{
26    public static void main(String args[]){
27        Student s1=new Student(111,"ankit","java");
28        Student s2=new Student(112,"sumit","java",6000);
29        s1.display();
30        s2.display();
31    }
32}
```

We will just want to discuss quickly one concept it is called the this keyword now this keyword concept can be understood very easily if we follow this example. Now let us consider the declaration of one class call the student. Here it this class has 4 member elements. Here integer roll number name and course as the String type and then key as the float.

Now, here again we see the this is the constructor declaration, roll number, name, String course and it. Now here you can note the arguments that we have passed they have their name and in this case the name of the data members are also same. Now if we try to initialize then definitely. So, for roll number equals to roll number, then this is not the correct similarly name equals to name it is also little bit ambiguous. So, what you can do is that we can we can define it by means of this one. So, this dot roll number indicates that this roll number belongs to this class member whereas, only roll number is the argument that has been passed similarly this dot name specify this the name members of the class Student and this dot course the name member of the class course.

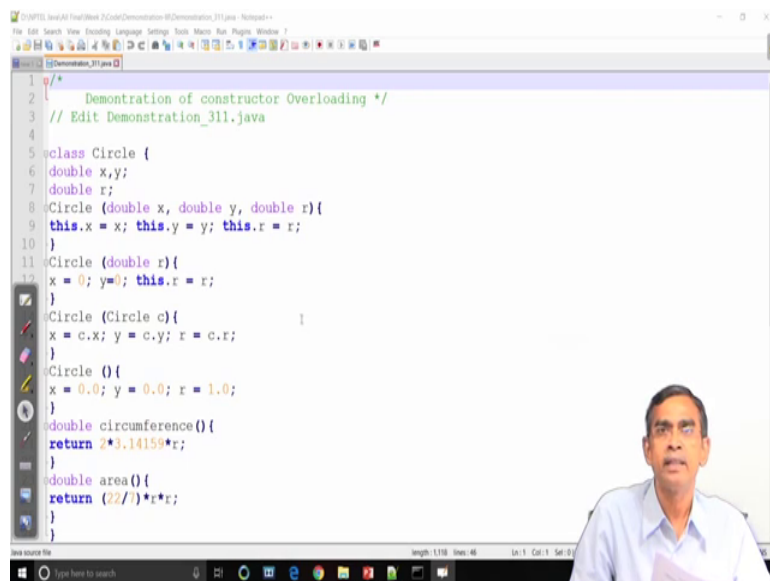
Now so now, if we run this one we will see exactly how we the next part will discuss latter on. Let us see how we can use it. So, this is basically this keyword which is basically helps to resolve the namespace collision. So, in this case roll number name and



course is been collided with the data member it is there. So, in order to resolve it we can use the this keyword. So, here we have used it and then we passed it.

Now, you can see the first time when the object is created first object on keep java 0.0 and now let us come to another way. Now this point we will discuss ok; let us come to the special use of this keyword in another 3.11 let us run the 3.11 demo.

(Refer Slide Time: 28:39)

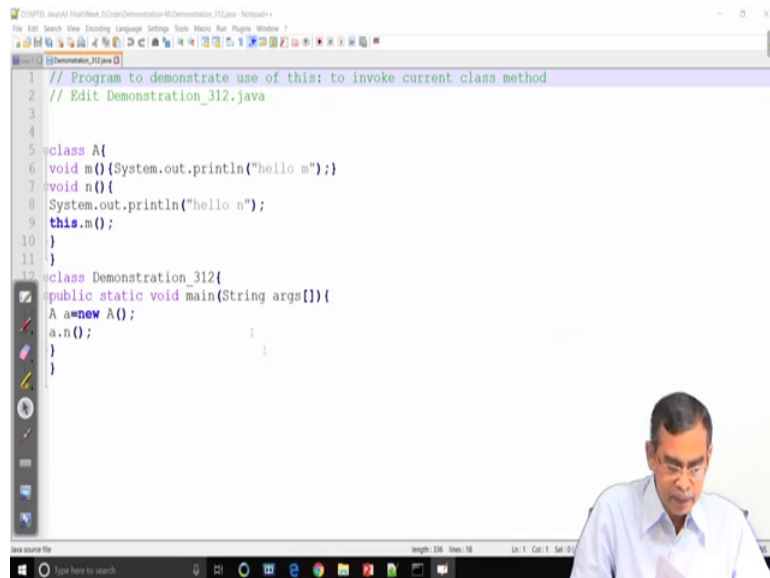


```
1 1 /*
2 2  Demonstration of constructor Overloading */
3 3  // Edit Demonstration_311.java
4 4
5 5 class Circle {
6 6  double x,y;
7 7  double r;
8 8  Circle (double x, double y, double r){
9 9  this.x = x; this.y = y; this.r = r;
10 10 }
11 11 Circle (double r){
12 12 x = 0; y=0; this.r = r;
13 13 }
14 14 Circle (Circle c){
15 15 x = c.x; y = c.y; r = c.r;
16 16 }
17 17 Circle (){
18 18 x = 0.0; y = 0.0; r = 1.0;
19 19 }
20 20 double circumference(){
21 21 return 2*3.14159*r;
22 22 }
23 23 double area(){
24 24 return (22/7)*r*r;
25 25 }
26 26 }
```

Now we will just discuss on more important thing here. Now in the context of Circle again the this keyword can be understood very precisely. So, here again constructor, we can see the constructor has been defined with x y r which have the same name as the member element and we resolve it by using this it is at the same thing applicable to the other also

So, if we want to resolve some name which belongs to the same class with other variable. So, we should use this. So, this is the one use of this keyword in java program. now there is another use of this program in the java keyword to understand this things let us have a simple program; let us have the program 3.12.

(Refer Slide Time: 29:23)



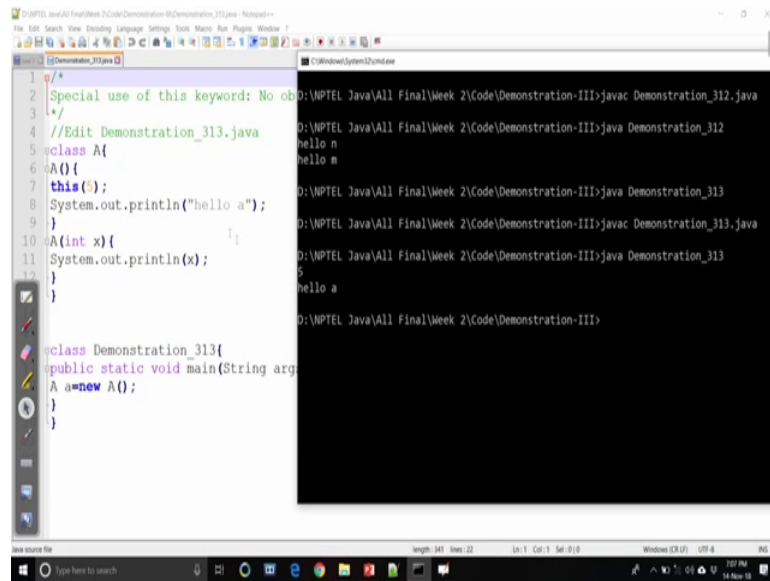
```
1 // Program to demonstrate use of this: to invoke current class method
2 // Edit Demonstration_312.java
3
4
5 class A{
6 void m(){System.out.println("hello m");}
7 void n(){
8 System.out.println("hello n");
9 this.m();
10 }
11 }
12
13 class Demonstration_312{
14 public static void main(String args[]){
15 A a=new A();
16 a.n();
17 }
18 }
```

Now, here we can see that this can be used to have some other advantage see. Here let us have program very simple one, it is very easy to understand. A is the class which declared here and it has one method m another method n the method m will simple print hello m, n also will print simple one String hello n.

Now, we if we want to access the same method in some other method, then definitely we have only learned that in order to access a method we should create an object. But whenever we are declaring a method or defining a method there is no question of creating any objects. Now if the objects belongs to the same; that means, if you want to access the method which is in the same class, then we can resolve it by specifying this dot m. For example, in the method void n we call the method m which is defined in this class itself.

So, in order to resolve it we used the this dot m this means that void n whenever we call for a new object created here for example, A then this will call the method m as well as method n. So, this so, so, when a dot n; that means, we call the method n; it will print hello n and then it will print that hello m. Now let us have the quick execution of this program so, that we can understand about this one.

(Refer Slide Time: 30:53)



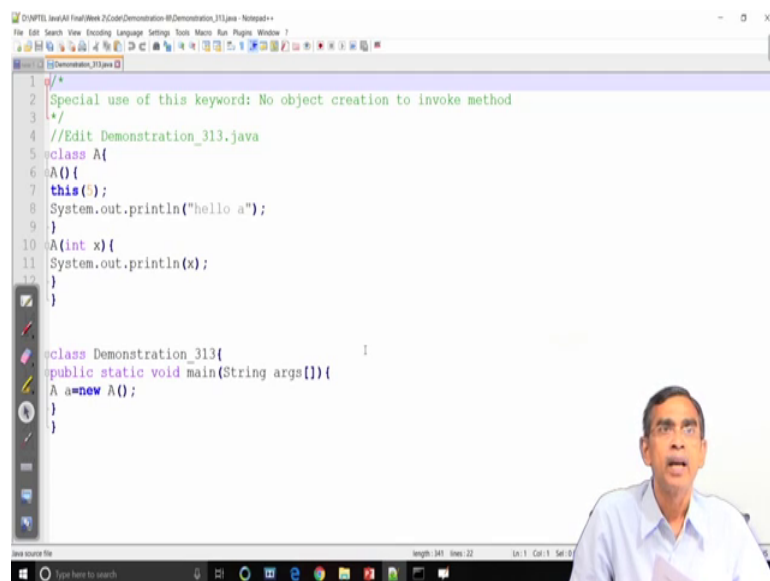
```
1 1 /*
2 2 Special use of this keyword: No ob
3 3 */
4 4 //Edit Demonstration_313.java
5 5 class A{
6 6 A(){
7 7 this();
8 8 System.out.println("hello a");
9 9 }
10 10 A(int x){
11 11 System.out.println(x);
12 12 }
13 13 }
14 14
15 15 class Demonstration_313{
16 16 public static void main(String arg
17 17 A a=new A();
18 18 }
19 19 }
```

Output:  
D:\NPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration\_312.java  
D:\NPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration\_312  
hello n  
hello m  
D:\NPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration\_313  
D:\NPTEL Java\All Final\Week 2\Code\Demonstration-III>javac Demonstration\_313.java  
D:\NPTEL Java\All Final\Week 2\Code\Demonstration-III>java Demonstration\_313  
hello a  
D:\NPTEL Java\All Final\Week 2\Code\Demonstration-III>

So, this method is used to refer one method which belongs to the same class without creating an object. Now this is the demonstration execution run of the program that we have discussed now ok.

We can understand. So, that the method m has been called from via n method which is defined now let us have the another quick example of this method it has many purpose that can be used who you want to give another.

(Refer Slide Time: 31:29)



```
1 1 /*
2 2 Special use of this keyword: No object creation to invoke method
3 3 */
4 4 //Edit Demonstration_313.java
5 5 class A{
6 6 A(){
7 7 this();
8 8 System.out.println("hello a");
9 9 }
10 10 A(int x){
11 11 System.out.println(x);
12 12 }
13 13 }
14 14
15 15 class Demonstration_313{
16 16 public static void main(String args[]){
17 17 A a=new A();
18 18 }
19 19 }
```

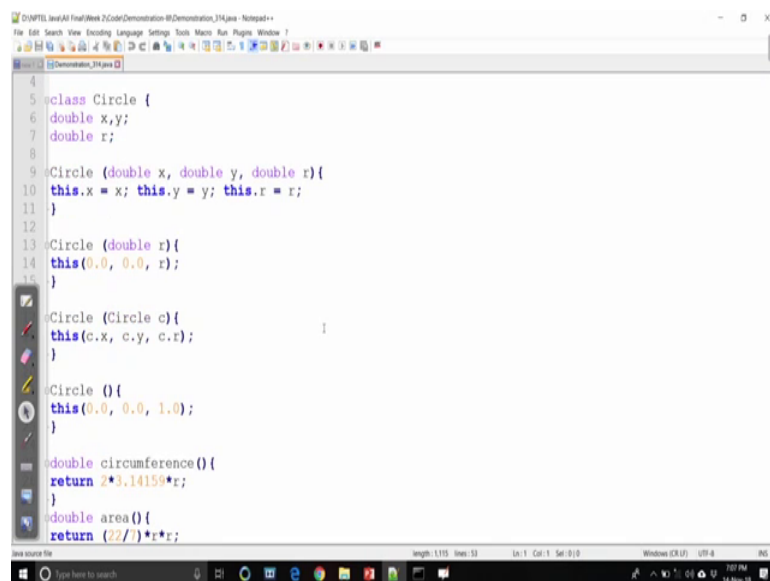
The screenshot shows the same Java code as the previous image. A video overlay of a presenter is visible in the bottom right corner of the IDE window.

So, this is the one method you see. So, class A is declared and A is the constructor go to the constructor definition there a constructor is there fine. We are clearing the A constructor; now here the constructor there are two constructors right the first constructor ok. Let us come to the second constructor what is the second constructor we define the second constructor by means of passing an argument then type integer. So, this means it will like this one right. Now here one default constructor which define who is basically initialized by a default value say 5.

But that installation calling by the one already defined constructor namely A intakes. So, if you want to call any constructor within another constructor, then you can use this and then specifying the parameter values that you want to have now let us have the execution of this program. So, this basically call a constructor in a constructor and this constructor is basically a nesting type of call it is it is also part of the overloading constructor. So, execution is there. So, you can see how the whenever we create an object here in this statement a equals to new A. We call the default constructor when first constructor whenever the first constructor is called it basically initialize the member elements with value 5. We have no member actually, the value is 5 is passed to the class object and then it print hello a using this value x.

So, basically it basically print as 5. So, this is the way. Now let us have a last day demo it is more interesting to understand.

(Refer Slide Time: 33:21)



```
4
5 class Circle {
6     double x,y;
7     double r;
8
9     Circle (double x, double y, double r){
10        this.x = x; this.y = y; this.r = r;
11    }
12
13    Circle (double r){
14        this(0.0, 0.0, r);
15    }
16
17    Circle (Circle c){
18        this(c.x, c.y, c.r);
19    }
20
21    Circle (){
22        this(0.0, 0.0, 1.0);
23    }
24
25    double circumference(){
26        return 2*3.14159*r;
27    }
28
29    double area(){
30        return (22/7)*r*r;
31    }
32 }
```

So, the last demo, it will be easier to understand. Now again Circle class. We create the one constructor is basically using 3 parameters. Now this constructor, then can be used to define other constructor by means of this the methods. So, circuit for example, second one this we call the same constructor here by using the this and passing 0.0, 0.0 r similar is the second constructor we use this c dot x c dot y c dot r. Here this and all these this basically indicates the one privilege constructor which is which has the three parameters and automatically it called them.

So, this way it basically has a little bit change in the similar program that we have discussed earlier that the these are use of these operator. So, we have discussed about constructor overloading and I just last things I just want to mention about. So, what the constructor overloading is concerned to constructors can be treated as a different; if they are different in terms of the argument number of arguments and the type of arguments. So, that is only things are there that you can already realize about it and if you can go through the all examples, then you will be able to understand that all three constructors or whatever the constructors they are different because of the different argument that we have passed ok. So, this concludes our demonstration on in encapsulation.

Thank you very much.