```java
// Polymorphism
program 1:
class OldBuilding
{

    void rooms()   // overriden method
    {
        System.out.println("2 rooms");
    }
}
class NewBuilding extends OldBuilding
{
    void rooms() // overriding method
    {
        System.out.println("5 rooms ");
    }
    public static void main(String[] args)
    {
        NewBuilding nb = new NewBuilding();
```

```
20          nb.rooms();
21      }
22
23  }
24  D:\AY 2023-24\SEM-I\JAVA\DIVB>java NewBuilding
25  5 rooms
26  ================================================
27  Program 2:
28  // Final class can't be overriden
29  final class OldBuilding
30  {
31
32      void rooms()   // overriden method
33      {
34          System.out.println("2 rooms");
35      }
36  }
37  class NewBuilding extends OldBuilding
38  {
```

```
39      void rooms() // overriding method
40      {
41          System.out.println("5 rooms ");
42      }
43      public static void main(String[] args)
44      {
45          NewBuilding nb = new NewBuilding();
46          nb.rooms();
47      }
48
49  }
50  D:\AY 2023-24\SEM-I\JAVA\DIVB>javac Polymorphism.java
51  Polymorphism.java:9: error: cannot inherit from final OldBuild
52  class NewBuilding extends OldBuilding
53                            ^
54  1 error
55  ================================================================
56  program 3
57  // final methods can't override
```

```
58 class OldBuilding
59 {
60
61     final void rooms()   // overriden method
62     {
63         System.out.println("2 rooms");
64     }
65 }
66 class NewBuilding extends OldBuilding
67 {
68     void rooms() // overriding method
69     {
70         System.out.println("5 rooms ");
71     }
72     public static void main(String[] args)
73     {
74         NewBuilding nb = new NewBuilding();
75         nb.rooms();
76     }
```

```
77
78 }
79
80 D:\AY 2023-24\SEM-I\JAVA\DIVB>javac Polymorphism.java
81 Polymorphism.java:11: error: rooms() in NewBuilding cannot ove
82         void rooms() // overriding method
83              ^
84   overridden method is final
85 1 error
86
87 ================================================================
88 program 4:
89 // return type of overriden and overriding method must be same
90 class OldBuilding
91 {
92
93      int rooms()   // overriden method
94     {
95         System.out.println("2 rooms");
```

```
 96          return 10;
 97      }
 98 }
 99 class NewBuilding extends OldBuilding
100 {
101     float rooms() // overriding method
102     {
103         System.out.println("5 rooms ");
104         return 5.5f;
105     }
106     public static void main(String[] args)
107     {
108         NewBuilding nb = new NewBuilding();
109         nb.rooms();
110     }
111 
112 }
113 D:\AY 2023-24\SEM-I\JAVA\DIVB>javac Polymorphism.java
114 Polymorphism.java:12: error: rooms() in NewBuilding cannot ove
```

```
115         float rooms() // overriding method
116                 ^
117   return type float is not compatible with int
118 1 error
119 ============================================================
120 program 5
121 // return type of overriden and overriding method may differ a
122 class MCA
123 {
124 }
125 class DIVB extends MCA
126 {
127 }
128
129 class OldBuilding
130 {
131
132      MCA rooms()   // overriden method
133     {
```

```
134             System.out.println("2 rooms");
135             return new MCA();
136         }
137 }
138 class NewBuilding extends OldBuilding
139 {
140         DIVB rooms() // overriding method
141         {
142             System.out.println("5 rooms ");
143             return new DIVB();
144         }
145         public static void main(String[] args)
146         {
147             NewBuilding nb = new NewBuilding();
148             nb.rooms();
149         }
150 
151 }
152 D:\AY 2023-24\SEM-I\JAVA\DIVB>java NewBuilding
```

```
5 rooms
==========================================================
program 6:
// final variables: cannot assign a value to final variable
class NewBuilding
{
    public static void main(String[] args)
    {
        final int a=100;
        a =a+10;
        System.out.println(a);


    }

}
D:\AY 2023-24\SEM-I\JAVA\DIVB>javac Polymorphism.java
Polymorphism.java:6: error: cannot assign a value to final var
                a =a+10;
```

```
                    ^
1 error
==========================================================
program 7:
// final class variables are not final but methods are final
final class NewBuilding
{
    int a=100; // not final
    void show() // final method
    {

        a =a+10;
        System.out.println(a);
    }
    public static void main(String[] args)
    {

        NewBuilding nb = new NewBuilding();
        nb.show();
```

```
191
192     }
193
194 }
195
196 ================================================
197 Program 8:
198 // type casting in polymorphism
199 class Parent
200 {
201     void m1() // overriden method
202     {
203         System.out.println("parent m1()");
204     }
205 }
206 class Child extends Parent
207 {
208     void m1() // overriding method
209     {
```

```
210         System.out.println("child m1()");
211     }
212     void m2() // direct method
213     {
214         System.out.println("Child m2()");
215     }
216 
217     public static void main(String [] args)
218     {
219         Parent  p = new Child();
220         p.m1();  // compile : parent  Runtime : child method
221         //p.m2();  // compile : parent error
222 
223         Child c = (Child) p; // type coversion /type casting
224         c.m1();
225         c.m2(); // compile : child
226 
227 
228     }
```

```
229
230 }
231 D:\AY 2023-24\SEM-I\JAVA\DIVB>java Child
232 child m1()
233 child m1()
234 Child m2()
235
236 ================================================================
237 program 9
238 // static methods can not override
239 class Parent
240 {
241     static void m1()
242     {
243         System.out.println("parent m1()");
244     }
245 }
246 class Child extends Parent
247 {
```

```
248     static void m1()
249     {
250         System.out.println("child m1()");
251     }
252     public static void main(String [] args)
253     {
254             Parent p = new Child();
255             p.m1();
256 
257     }
258 
259 }
260 D:\AY 2023-24\SEM-I\JAVA\DIVB>java Child
261 parent m1()
262 ================================================
263 Program 10
264 // private methods can not override in child class
265 class Parent
266 {
```

```
267     private void m1()
268     {
269         System.out.println("parent m1()");
270     }
271 }
272 class Child extends Parent
273 {
274     private void m1()
275     {
276         System.out.println("child m1()");
277     }
278     public static void main(String [] args)
279     {
280             Parent  p = new Parent();
281             p.m1();
282 
283     }
284 
285 }
```

```
D:\AY 2023-24\SEM-I\JAVA\DIVB>javac Polymorphism.java
Polymorphism.java:17: error: m1() has private access in Parent
                          p.m1();
                           ^
1 error
==================================================
program 11
// same level permissions  ex: default

class Parent
{
    void m1()  // overriden method
    {
        System.out.println("parent m1()");
    }
}
class Child extends Parent
{
    void m1() //overriding method
```

```
305     {
306         System.out.println("child m1()");
307     }
308 }
309 
310 ================================================
311 program 12
312 // permissions are decreses here means public to default
313 class Parent
314 {
315     public void m1()  // overriden method
316     {
317         System.out.println("parent m1()");
318     }
319 }
320 class Child extends Parent
321 {
322     void m1() //overriding method
323     {
```

```
324         System.out.println("child m1()");
325     }
326 }
327 D:\AY 2023-24\SEM-I\JAVA\DIVB>javac Polymorphism.java
328 Polymorphism.java:10: error: m1() in Child cannot override m1(
329         void m1() //overriding method
330              ^
331   attempting to assign weaker access privileges; was public
332 1 error
333 ================================================================
334 program 13
335 // permission increases  ,which is allowed in overriding
336 class Parent
337 {
338     void m1()  // overriden method
339     {
340         System.out.println("parent m1()");
341     }
342 }
```

```
343 class Child extends Parent
344 {
345     protected void m1() //overriding method
346     {
347         System.out.println("child m1()");
348     }
349 }
350
351
352
353
354
355
356
357
```