

```

1 package com.publiccept;
2
3 /**
4  * Main
5  * main(String[] args)
6  *
7  * randomMovie()
8  */
9 public class Main {
10
11     /**
12      * main()
13      * @param args args
14      */
15     public static void main(String[] args) {
16
17         /**
18          * Create polymorphic variable "movie", containing different shapes/types (=(sub-)classes) of Movie,
19          * while all of them really are different shapes/types of "Movie".
20          */
21         for (int i=0; i<10; i++) {
22
23             Movie movie = randomMovie();
24             System.out.println("Movie #" + i + " : " + movie.getName() + "\n" +
25                             "Plot: " + movie.plot() + "\n");
26         }
27     }
28
29
30     /**
31      * randomMovie()
32      * @return random movie
33      */
34     public static Movie randomMovie() {
35
36         // Math.random() = returns a random double value between 0 and 1
37         int randomNumber = (int) (Math.random() * 6); // generate random number from 0 to 6
38         System.out.println("random number generated was: " + randomNumber);
39
40         switch (randomNumber) {
41             case 0:
42                 // =====
43                 // == Super/parent object showcase ==
44                 // =====
45                 return new Movie("Empty dummy movie"); // showcase of a super/parent object created
46             case 1:
47                 return new Jaws();
48             case 2:
49                 return new IndependenceDay();
50             case 3:
49                 return new MazeRunner();
51             case 4:
52                 return new StarWars();
53             case 5:
54                 // =====
55                 // == Polymorphism showcase ==
56                 // =====
57                 return new Forgettable(); // showcase of a different sub/child Movie object created
58             default:
59                 return null;
60         }
61     }
62 }
63
64
65
66
67 /**
68  * Movie
69  * Movie(String name)
70  *
71  * plot()
72  * getName()
73  */
74 class Movie {
75
76     // == fields ==
77     private String name;
78
79     // == constructors ==
80     public Movie(String name) {
81         this.name = name;
82     }
83
84     // == public methods ==
85     public String plot() {
86         return "No plot here";
87     }
88
89     public String getName() {
90         return name;
91     }
92 }
93
94
95
96 /**
97  * Jaws extends Movie

```

```

98  * Jaws()
99  *
100 * plot()
101 */
102 class Jaws extends Movie {
103
104     public Jaws() {
105         super("Jaws");
106     }
107
108     @Override
109     public String plot() {
110         return "A shark eats lots of people";
111     }
112 }
113
114
115
116 /**
117  * IndependenceDay extends Movie
118  * IndependenceDay()
119  *
120  * plot()
121  */
122 class IndependenceDay extends Movie {
123
124     public IndependenceDay() {
125         super("Independence Day");
126     }
127
128     @Override
129     public String plot() {
130         return "Aliens attempt to take over planet earth";
131     }
132 }
133
134
135
136 /**
137  * MazeRunner extends Movie
138  * MazeRunner()
139  *
140  * plot()
141  */
142 class MazeRunner extends Movie {
143
144     public MazeRunner() {
145         super("Maze Runner");
146     }
147
148     @Override
149     public String plot() {
150         return "Kids try and escape a maze";
151     }
152 }
153
154
155
156 /**
157  * StarWars extends Movie
158  * StarWars()
159  *
160  * plot()
161  */
162 class StarWars extends Movie {
163
164     public StarWars() {
165         super("Star Wars");
166     }
167
168     @Override
169     public String plot() {
170         return "Imperial forces try to take over the universe";
171     }
172 }
173
174
175
176 /**
177  * Forgettable extends Movie
178  * Forgettable()
179  *
180  * =====
181  * == Polymorphism showcase ==
182  * =====
183  * super class plot is shown if #5 Forgettable is executed
184  */
185 class Forgettable extends Movie {
186
187     public Forgettable() {
188         super("Forgettable");
189     }
190
191     // =====
192     // == Polymorphism showcase ==
193     // =====
194     // No plot method implemented:
195     // returns "No plot here" from the super Movie class

```

