

```

1  /**
2   * Main
3   */
4  public class Main {
5
6      public static void main(String[] args) {
7
8          int newScore = Score.calculateScore("Urs", 500);
9          System.out.println("New score is " + newScore);
10         Score.calculateScore(75);
11         Score.calculateScore();
12
13
14         /**
15          * Challenge: Calculation of centimeters from feet and inches
16          */
17         // Create a method called calcFeetAndInchesToCentimeters
18         // It needs to have two parameters.
19         // feet is the first parameter, inches is the 2nd parameter
20         //
21         // You should validate that the first parameter feet is >= 0
22         // You should validate that the 2nd parameter inches is >=0 and <=12
23         // return -1 from the method if either of the above is not true
24         //
25         // If the parameters are valid, then calculate how many centimetres
26         // comprise the feet and inches passed to this method and return
27         // that value.
28         //
29         // Create a 2nd method of the same name but with only one parameter
30         // inches is the parameter
31         // validate that its >=0
32         // return -1 if it is not true
33         // But if its valid, then calculate how many feet are in the inches
34         // and then here is the tricky part
35         // call the other overloaded method passing the correct feet and inches
36         // calculated so that it can calculate correctly.
37         // hints: Use double for your number datatypes is probably a good idea
38         // 1 inch = 2.54cm and one foot = 12 inches
39         // use the link I give you to confirm your code is calculating correctly.
40         // Calling another overloaded method just requires you to use the
41         // right number of parameters.
42
43         double feet = 1.00d;
44         double inches = 1.00d;
45
46         double centimeters = CentimetersFromFeet.calcFeetAndInchesToCentimeters(feet, inches);
47         if (centimeters < 0.0) {
48             System.out.println("invalid parameters");
49         }
50
51         inches = 100.00d;
52         centimeters = CentimetersFromFeet.calcFeetAndInchesToCentimeters(inches);
53         if (centimeters < 0.00) {
54             System.out.println("invalid parameters");
55         }
56     }
57 }
58
59
60
61 /**
62  * CentimetersFromFeet
63  *
64  * calcFeetAndInchesToCentimeters(double inches)
65  * calcFeetAndInchesToCentimeters(double feet, double inches)
66  */
67 class CentimetersFromFeet {
68
69
70     /**
71      * Calculate centimeters from inches(one param only)
72      * @param inches
73      * @return
74      */
75     public static double calcFeetAndInchesToCentimeters(double inches) {
76         if (inches >= 0) {
77             double feet = (int) inches / 12; // (cast double to int)
78             double remainingInches = (int) inches % 12.00d; // (cast double to int)
79             System.out.println(inches + " inches is equal to " + feet + " feet and " + remainingInches + " inches.");
80             return calcFeetAndInchesToCentimeters(feet, remainingInches);
81         }
82         return -1.00d;
83     }
84
85
86     /**
87      * calcFeetAndInchesToCentimeters(two params - overloaded)
88      *
89      * @param feet feet
90      * @param inches inches
91      * @return cm
92      */
93     public static double calcFeetAndInchesToCentimeters(double feet, double inches) {
94         if (feet >= 0 && (inches >= 0 && inches < 12)) {
95             inches += 12 * feet;
96             double centimeters = 2.54 * inches;
97             System.out.println(feet + " ft + " + inches + " in = " + centimeters + " cm");

```

```

98         return centimeters;
99     } else {
100         return -1.00d;
101     }
102 }
103 }
104
105
106
107 /**
108  * Score
109  *
110  * calculateScore()
111  * calculateScore(int score)
112  * calculateScore(String playerName, int score)
113  */
114 class Score {
115
116
117     /**
118      * Calculate Score (2 params)
119      * @param playerName
120      * @param score
121      * @return
122      */
123     // method signature: "calculateScore(String playerName, int score)" = name and parameters
124     public static int calculateScore(String playerName, int score) {
125         System.out.println("Player " + playerName + " scored " + score + " points");
126         return score * 1000;
127     }
128
129
130     /**
131      * Calculate Score (1 Param)
132      * @param score
133      * @return
134      */
135     public static int calculateScore(int score) {
136         System.out.println("Unnamed player scored " + score + " points");
137         return score * 1000;
138     }
139
140
141     /**
142      * Calculate Score (no params)
143      * @return
144      */
145     public static int calculateScore() {
146         System.out.println("No player name, no player score");
147         return 0;
148     }
149 }
150
151
152

```