```
1  //
2  // Meter class source file for A5
3  //
4  // Hal Bettle
5  //
6  // 5 September 2008
7  //
8  #include "CPTN230A5class_meter_bettle.h"
9
10 Meter::Meter()
11 {
12     cout << "Meter Default Constructor." << endl;
13     meters = 0;
14 }
15
16 Meter::Meter(double d)
17 {
18     cout << "Meter Double Conversion Constructor." << endl;
19     meters = d;
20 }
21
22 Meter::Meter(Meter &M)
23 {
24     cout << "Meter Copy Constructor." << endl;
25     meters = M.meters;
26 }
27
28 Meter::Meter(Foot &F)
29 {
30     cout << "Meter Foot Conversion Constructor." << endl;
31     meters = F.feet / 3.281;
32 }
33
34 Meter::~Meter()
35 {
36     cout << "Meter Default Destructor." << endl;
37 }
38
39 double Meter::get_meters(void)
40 {
41     return meters;
42 }
43
44 void Meter::operator=(Meter &M)
45 {
46     cout << "Meter Default Assignment Operator." << endl;
47     meters = M.meters;
48     return;
49 }
50
51 void Meter::operator=(double d)
52 {
53     cout << "Meter Double Assignment Operator." << endl;
54     meters = d;
55     return;
56 }
57
58 void Meter::operator=(Foot &F)
59 {
60     cout << "Meter Foot Assignment Operator." << endl;
61     meters = F.feet / 3.281;
62     return;
63 }
64
65 Meter Meter::operator-()
66 {
```

```
 67     Meter temp;
 68     cout << "Meter Negation Operator." << endl;
 69     temp.meters = -meters;
 70     return temp;
 71 }
 72
 73 Meter Meter::operator+(const Meter &M)
 74 {
 75     cout << "Meter Meter + Meter Operator." << endl;
 76     return Meter( meters + M.meters);
 77 }
 78
 79 Meter Meter::operator-(const Meter &M)
 80 {
 81     cout << "Meter Meter - Meter Operator." << endl;
 82     return Meter( meters - M.meters);
 83 }
 84
 85 Meter Meter::operator*(const Meter &M)
 86 {
 87     cout << "Meter Meter * Meter Operator." << endl;
 88     return Meter( meters * M.meters);
 89 }
 90
 91 Meter Meter::operator/(const Meter &M)
 92 {
 93     cout << "Meter Meter / Meter Operator." << endl;
 94     if (M.meters == 0)
 95     {
 96         cout << "Attempt to divide by 0." << endl;
 97         return Meter(0);
 98     }
 99     else
100     {
101         return Meter( meters / M.meters);
102     }
103 }
104
105 Meter Meter::operator+(const Foot &F)
106 {
107     cout << "Meter Meter + Foot Operator." << endl;
108     return Meter( meters + (F.feet / 3.281));
109 }
110
111 Meter::operator Foot()
112 {
113     cout << "Meter Foot Cast Operator." << endl;
114     return Foot(meters * 3.281);
115 }
```