```
1  //
2  // Foot class source file for A5
3  //
4  // Hal Bettle
5  //
6  // 5 September 2008
7  //
8  #include "CPTN230A5class_foot_bettle.h"
9
10 Foot::Foot()
11 {
12     cout << "Foot Default Constructor." << endl;
13     feet = 0;
14 }
15
16 Foot::Foot(double d)
17 {
18     cout << "Foot Double Conversion Constructor." << endl;
19     feet = d;
20 }
21
22 Foot::Foot(Foot &F)
23 {
24     cout << "Foot Copy Constructor." << endl;
25     feet = F.feet;
26 }
27
28 Foot::Foot(Meter &M)
29 {
30     cout << "Foot Meter Conversion Constructor." << endl;
31     feet = M.meters * 3.281;
32 }
33
34 Foot::~Foot()
35 {
36     cout << "Foot Default Destructor." << endl;
37 }
38
39 double Foot::get_feet(void)
40 {
41     return feet;
42 }
43
44 void Foot::operator=(Foot &F)
45 {
46     cout << "Foot Default Assignment Operator." << endl;
47     feet = F.feet;
48     return;
49 }
50
51 void Foot::operator=(double d)
52 {
53     cout << "Foot Double Assignment Operator." << endl;
54     feet = d;
55     return;
56 }
57
58 void Foot::operator=(Meter &M)
59 {
60     cout << "Foot Meter Assignment Operator." << endl;
61     feet = M.meters * 3.281;
62     return;
63 }
64
65 Foot Foot::operator-()
66 {
```

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