MA122 - Computer Programming and Apllications

Indian Institute of Space Science and Technology

March 02, 2017

Lecture 17

1 Pointer to a function

Pointer to a function


```
1  #include <iostream>
2  double betsy(int);
3  double pam(int);
4  void estimate(int lines, double (*pf)(int));
5  int main()
6  {
7    using namespace std;
8    int code;
9    cout << "How many lines of code do you need? ";
10   cin >> code;
11   cout << "Here's Betsy's estimate:\n";
12   estimate(code, betsy);
13   cout << "Here's Pam's estimate:\n";
14   estimate(code, pam);
15   return 0;
16 }
```

Pointers to a Function


```
1  double betsy(int lns)
2  {
3    return 0.05 * lns;
4  }
5  double pam(int lns)
6  {
7    return 0.03 * lns + 0.0004 * lns * lns;
8  }
9  void estimate(int lines, double (*pf)(int))
10 {
11   using namespace std;
12   cout << lines << " lines will take ";
13   cout <<(*pf)(lines) << " hour(s)\n";
14 }
```

Pointer to a function


```
1  #include <iostream>
2  double square(double y);
3  int main()
4  {
5      double (*foo)(double);
6      double x;
7      foo=&square;
8      x=foo(1.1);
9      std::cout<<x<<std::endl;
10     x=(*foo)(1.1);
11     std::cout<<x<<std::endl;
12     return 0;
13 }
14 double square(double y)
15 {
16     return y*y;
17 }
```

Pointer to a function


```
1  #include <iostream>
2  int* DoubleValue(int nX);
3  int main()
4  {
5    using namespace std;
6
7    cout<<DoubleValue(5)<<endl;
8
9    return 0;
10 }
11 int *DoubleValue(int nX)
12 {
13   int nValue=nX*2;
14   int *pt3;
15   pt3=&nValue;
16   std::cout<<&nValue<<std::endl;
17   return &nValue;
18 }
```

Function to a function


```
1  // pointer to functions
2  #include <iostream>
3  using namespace std;
4  int addition(int a, int b);
5  int subtraction(int a, int b);
6  int operation(int x, int y, int (*functioncall)(int,
       int));
7  int main ()
8  {
9      int m,n;
10     int (*minus)(int,int) = subtraction;
11
12     m = operation (7, 5, addition);
13     n = operation (20, m, minus);
14     cout <<n<<endl;
15     return 0;
16 }
```

Function to a Function


```
1  int addition (int a, int b)
2  { return (a+b); }
3
4  int subtraction (int a, int b)
5  { return (a-b); }
6
7  int operation (int x, int y, int (*functioncall)(int,
       int))
8  {
9      int g;
10     g = (*functioncall)(x,y);
11     return (g);
12 }
```