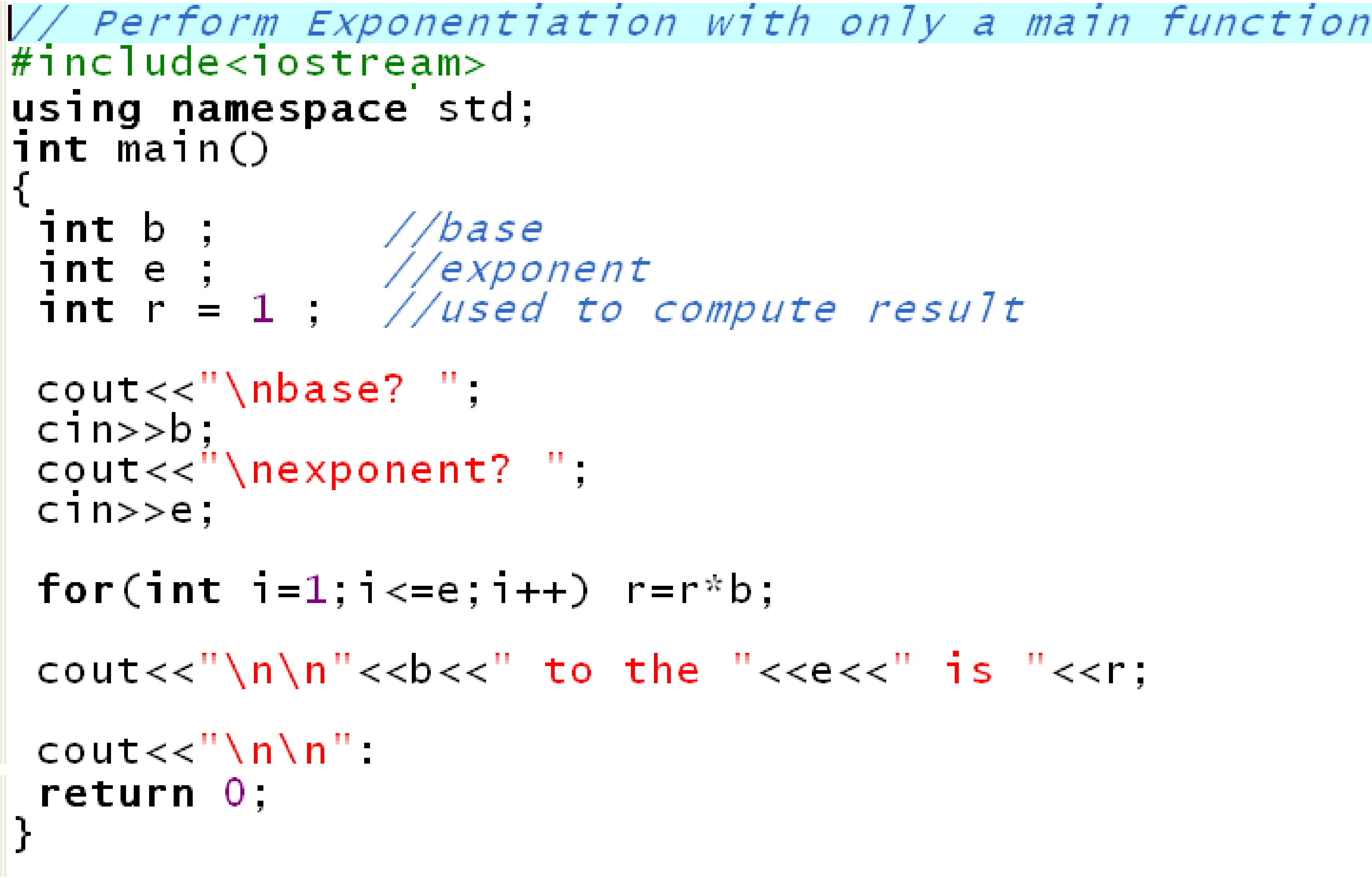
C++
Functions:

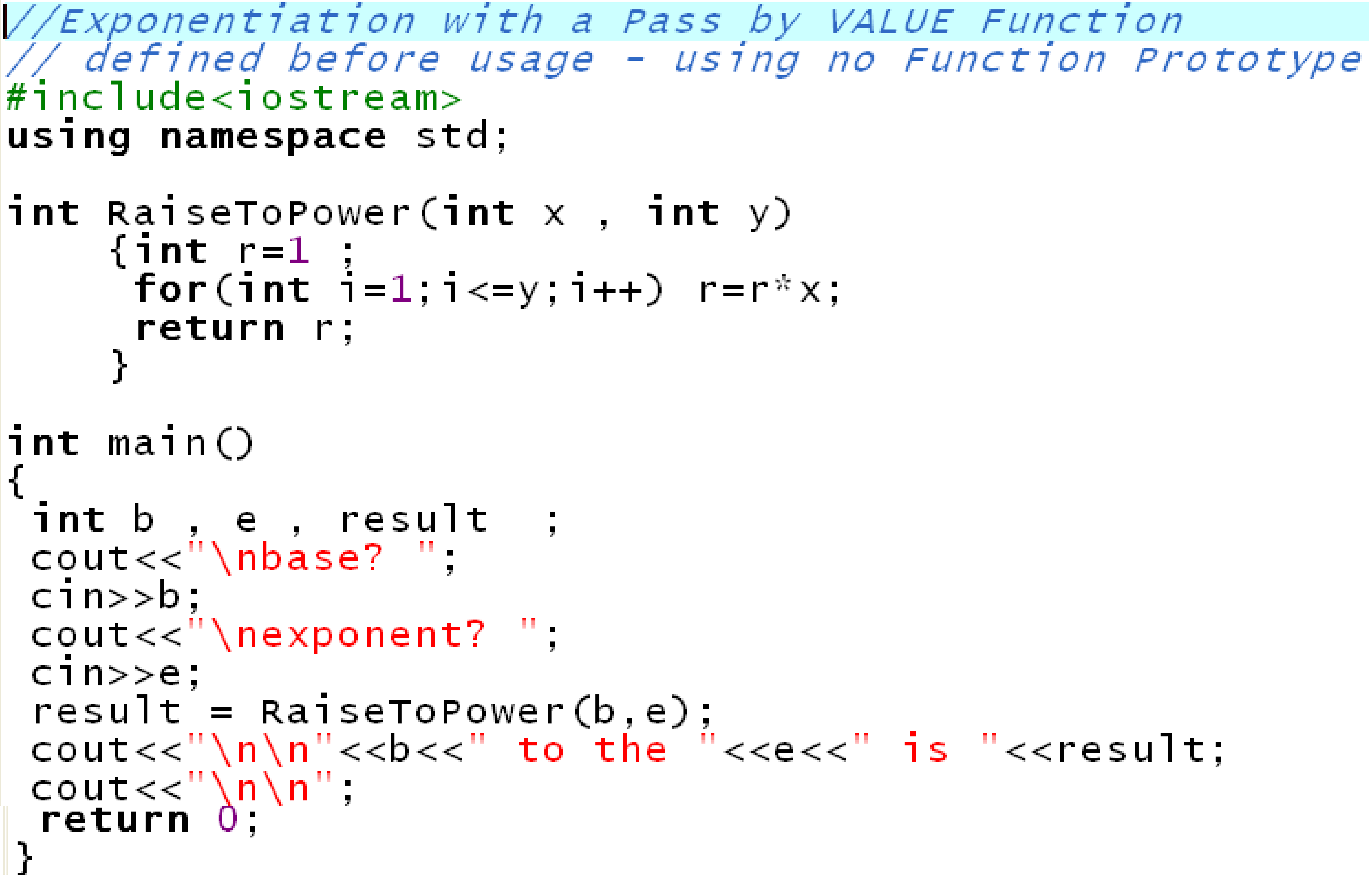
many options and techniques are available to the C++ programmer

Kingsborough Computer Science

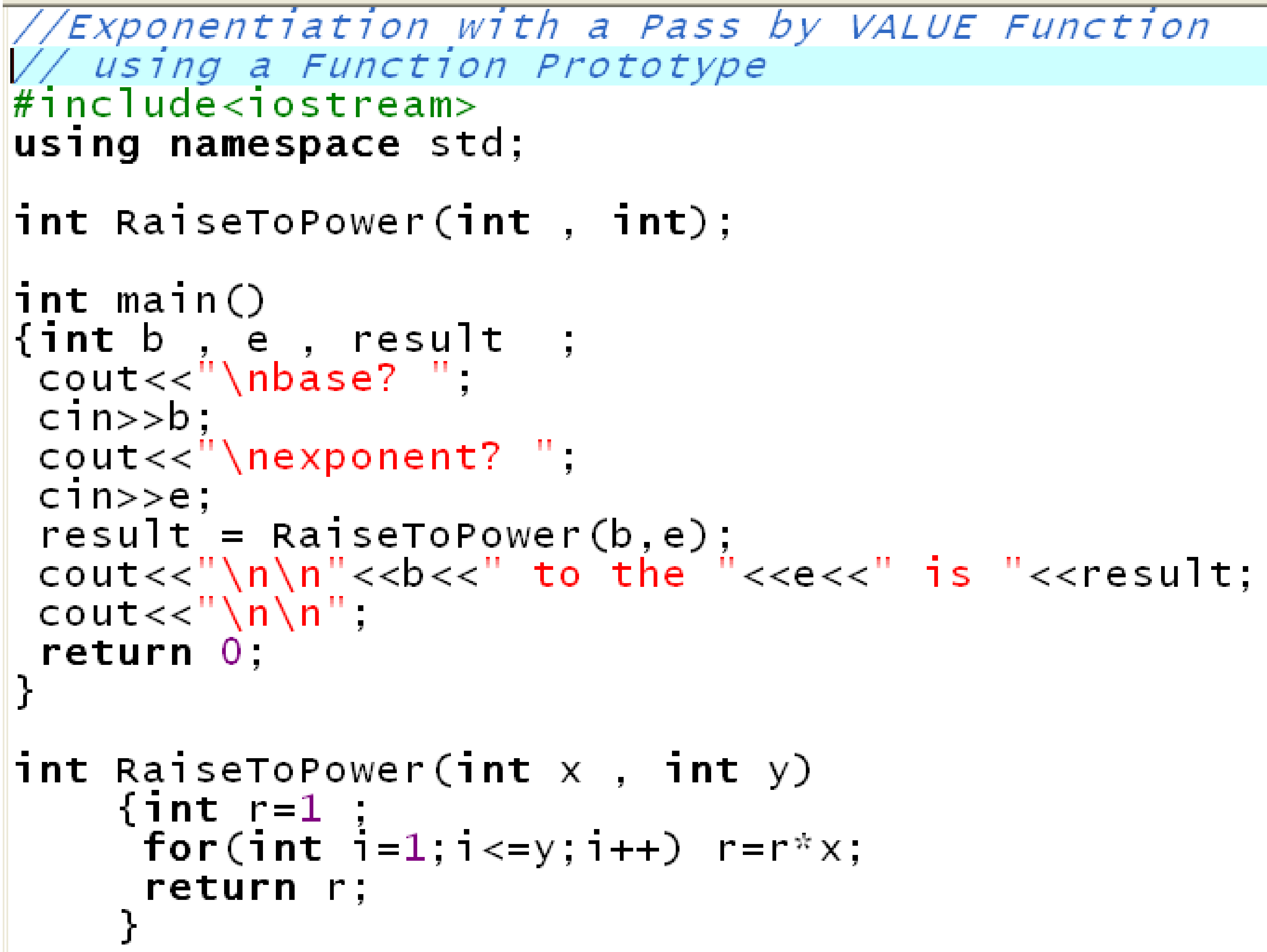
Dr. Philip Listowsky



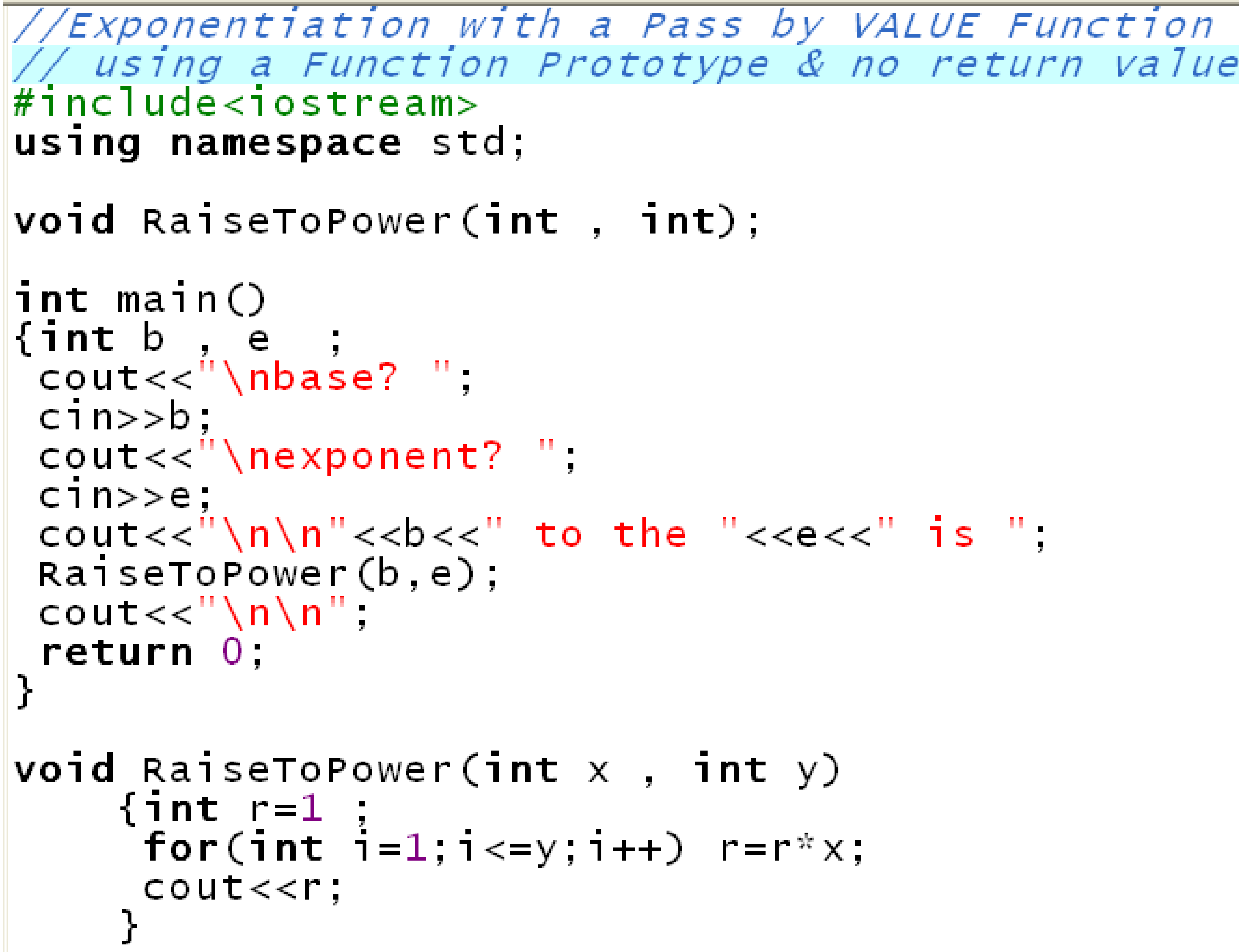
```cpp
// Perform Exponentiation with only a main function
#include<iostream>
using namespace std;
int main()
{
 int b ;      //base
 int e ;      //exponent
 int r = 1 ;  //used to compute result

 cout<<"\nbase? ";
 cin>>b;
 cout<<"\nexponent? ";
 cin>>e;

 for(int i=1;i<=e;i++) r=r*b;

 cout<<"\n\n"<<b<<" to the "<<e<<" is "<<r;

 cout<<"\n\n":
 return 0;
}
```



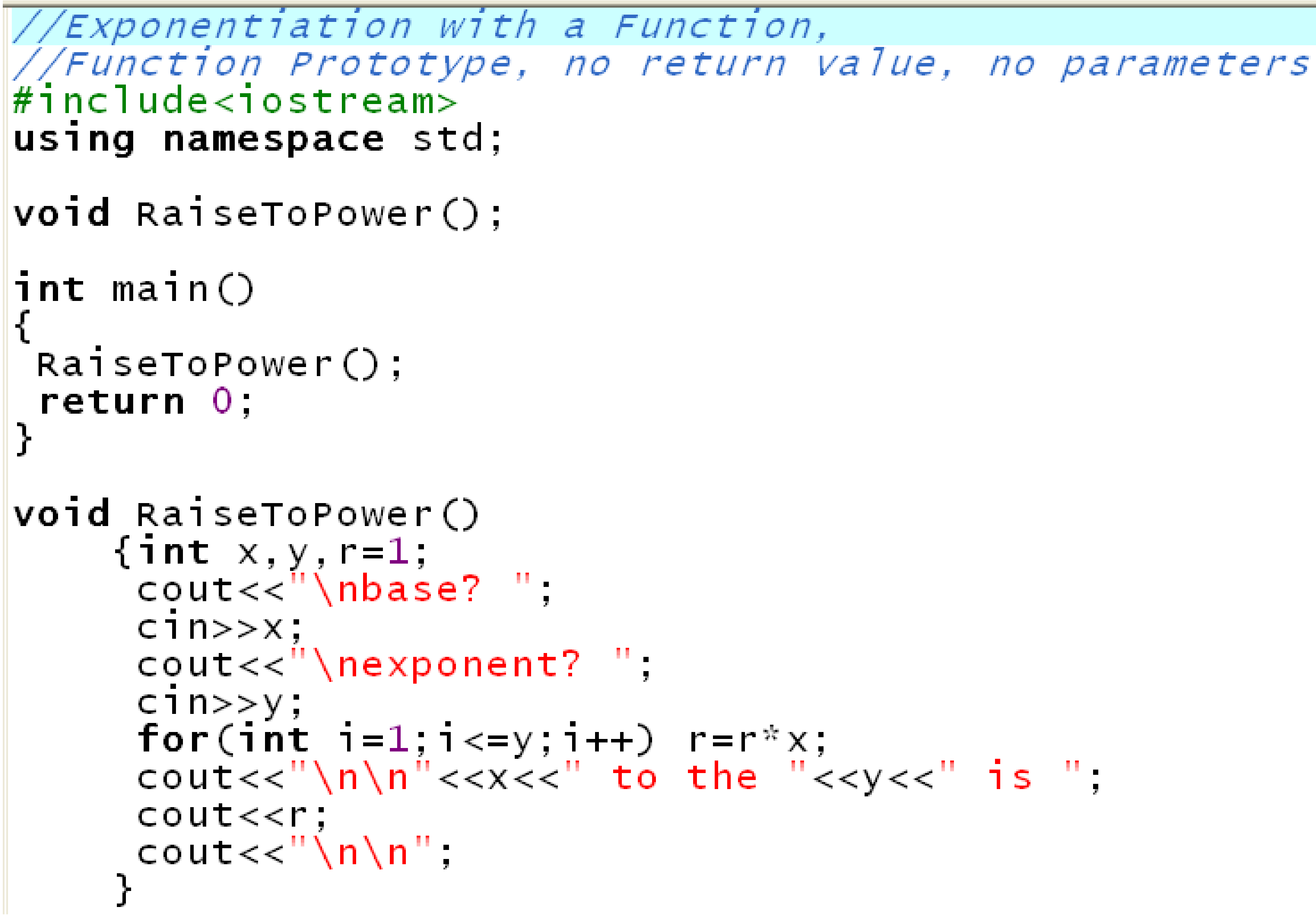
```cpp
//Exponentiation with a Pass by VALUE Function
// defined before usage - using no Function Prototype
#include<iostream>
using namespace std;

int RaiseToPower(int x , int y)
    {int r=1 ;
     for(int i=1;i<=y;i++) r=r*x;
     return r;
    }

int main()
{
 int b , e , result  ;
 cout<<"\nbase? ";
 cin>>b;
 cout<<"\nexponent? ";
 cin>>e;
 result = RaiseToPower(b,e);
 cout<<"\n\n"<<b<<" to the "<<e<<" is "<<result;
 cout<<"\n\n";
 return 0;
}
```



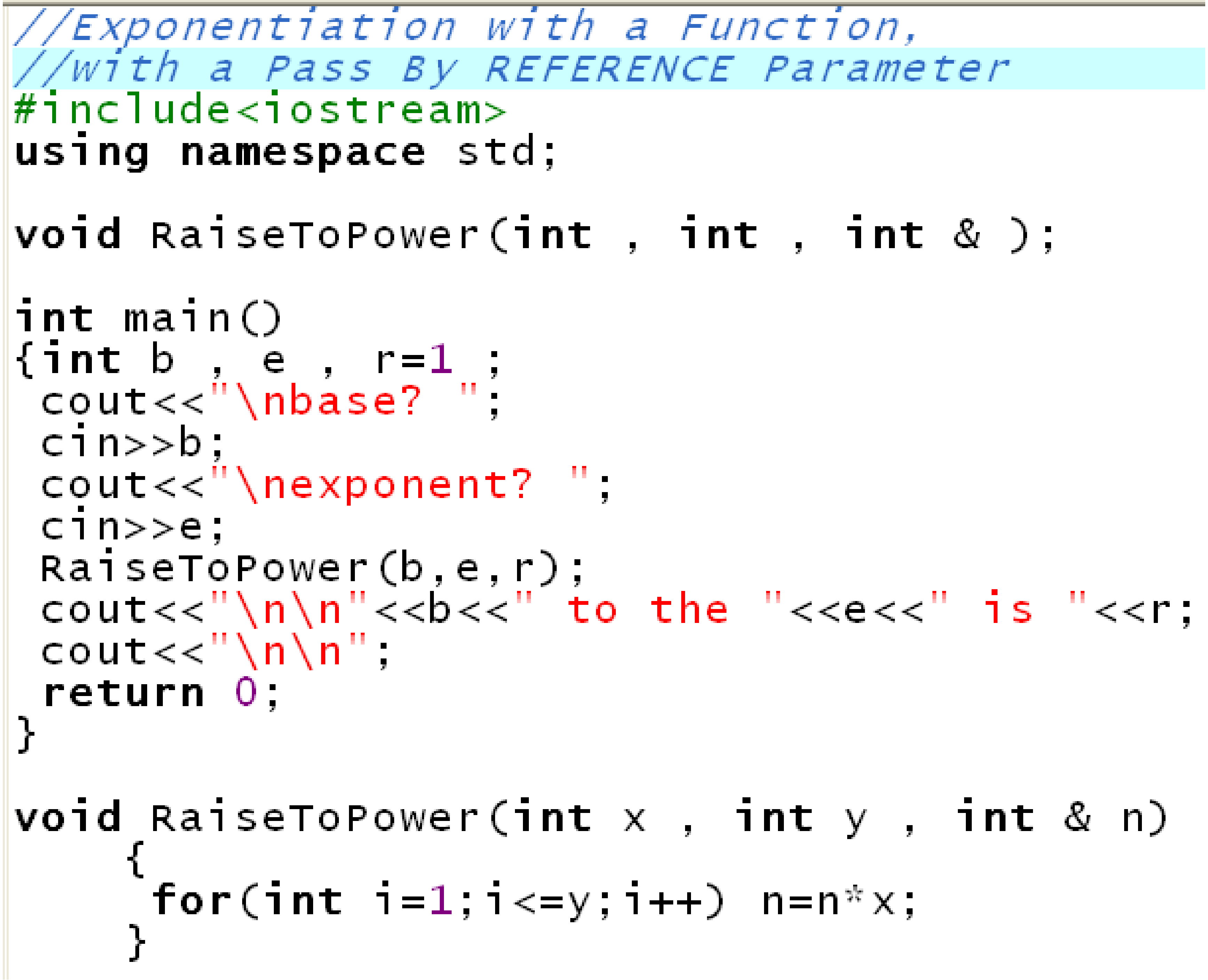
```
//Exponentiation with a Pass by VALUE Function
// using a Function Prototype
#include<iostream>
using namespace std;

int RaiseToPower(int , int);

int main()
{int b , e , result  ;
 cout<<"\nbase? ";
 cin>>b;
 cout<<"\nexponent? ";
 cin>>e;
 result = RaiseToPower(b,e);
 cout<<"\n\n"<<b<<" to the "<<e<<" is "<<result;
 cout<<"\n\n";
 return 0;
}

int RaiseToPower(int x , int y)
    {int r=1 ;
     for(int i=1;i<=y;i++) r=r*x;
     return r;
    }
```



```cpp
//Exponentiation with a Pass by VALUE Function
// using a Function Prototype & no return value
#include<iostream>
using namespace std;

void RaiseToPower(int , int);

int main()
{int b , e  ;
 cout<<"\nbase? ";
 cin>>b;
 cout<<"\nexponent? ";
 cin>>e;
 cout<<"\n\n"<<b<<" to the "<<e<<" is ";
 RaiseToPower(b,e);
 cout<<"\n\n";
 return 0;
}

void RaiseToPower(int x , int y)
    {int r=1 ;
     for(int i=1;i<=y;i++) r=r*x;
     cout<<r;
    }
```

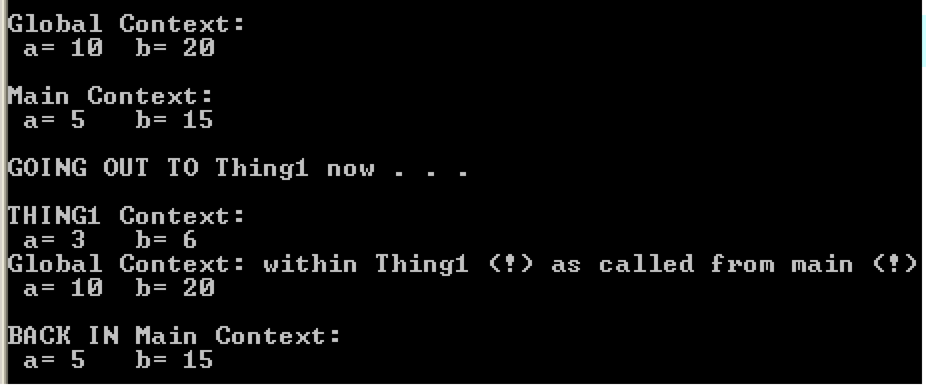


```
//Exponentiation with a Function,
//Function Prototype, no return value, no parameters
#include<iostream>
using namespace std;

void RaiseToPower();

int main()
{
 RaiseToPower();
 return 0;
}

void RaiseToPower()
    {int x,y,r=1;
     cout<<"\nbase? ";
     cin>>x;
     cout<<"\nexponent? ";
     cin>>y;
     for(int i=1;i<=y;i++) r=r*x;
     cout<<"\n\n"<<x<<" to the "<<y<<" is ";
     cout<<r;
     cout<<"\n\n";
    }
```

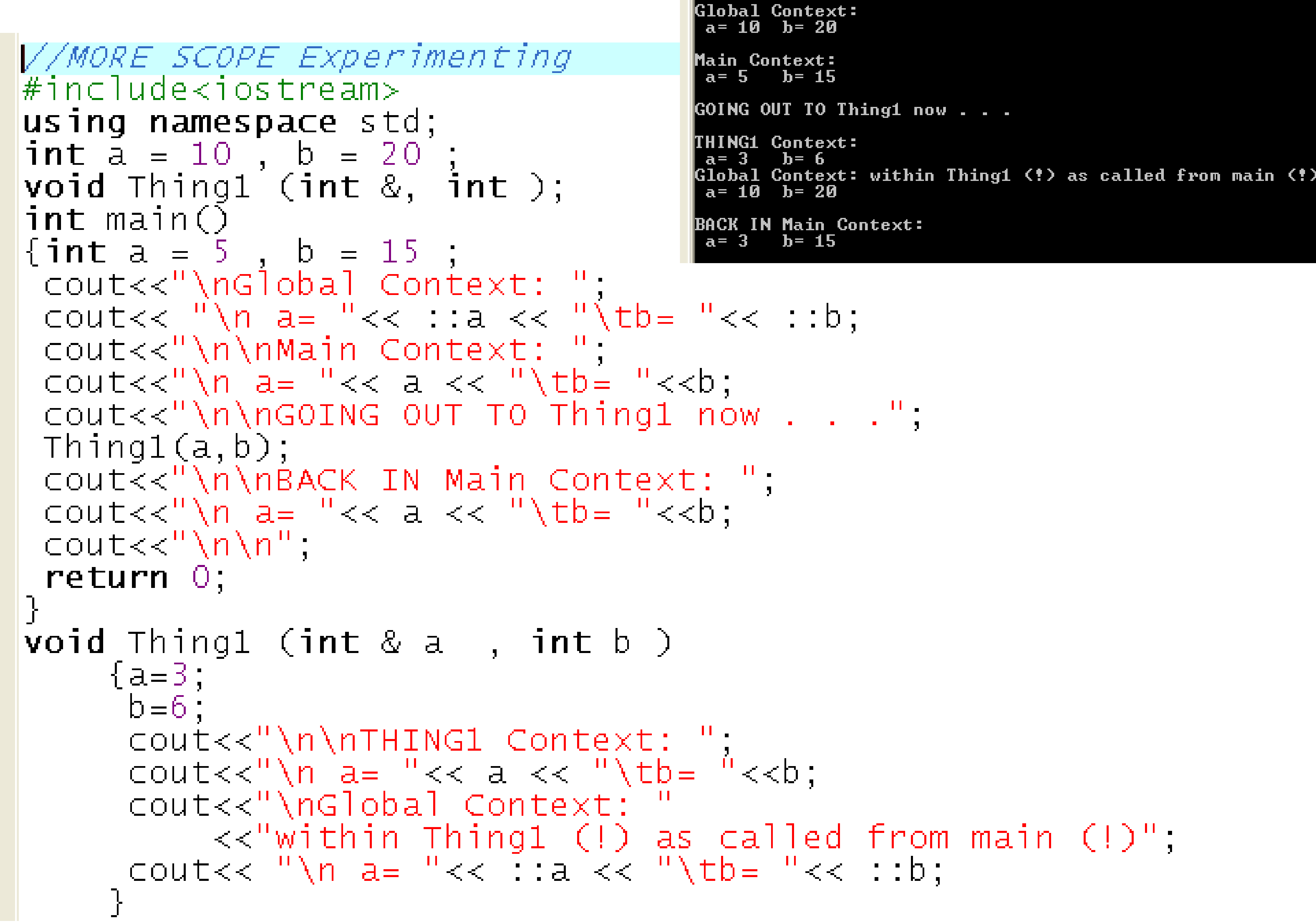


```
//Exponentiation with a Function,
//with a Pass By REFERENCE Parameter
#include<iostream>
using namespace std;

void RaiseToPower(int , int , int & );

int main()
{int b , e , r=1 ;
 cout<<"\nbase? ";
 cin>>b;
 cout<<"\nexponent? ";
 cin>>e;
 RaiseToPower(b,e,r);
 cout<<"\n\n"<<b<<" to the "<<e<<" is "<<r;
 cout<<"\n\n";
 return 0;
}

void RaiseToPower(int x , int y , int & n)
    {
     for(int i=1;i<=y;i++) n=n*x;
    }
```

```cpp
//Experimenting with SCOPE
#include<iostream>
using namespace std;
int a = 10 , b = 20 ;
void Thing1 (int , int);
int main()
{int a = 5 , b = 15 ;
 cout<<"\nGlobal Context: ";
 cout<< "\n a= "<< ::a << "\tb= "<< ::b;
 cout<<"\n\nMain Context: ";
 cout<<"\n a= "<< a << "\tb= "<<b;
 cout<<"\n\nGOING OUT TO Thing1 now . . .";
 Thing1(a,b);
 cout<<"\n\nBACK IN Main Context: ";
 cout<<"\n a= "<< a << "\tb= "<<b;
 cout<<"\n\n";
 return 0;
}
void Thing1 (int a , int b )
    {a=3;
     b=6;
     cout<<"\n\nTHING1 Context: ";
     cout<<"\n a= "<< a << "\tb= "<<b;
     cout<<"\nGlobal Context: "
         <<"within Thing1 (!) as called from main (!)";
     cout<< "\n a= "<< ::a << "\tb= "<< ::b;
    }
```



```
Global Context:
 a= 10  b= 20

Main Context:
 a= 5   b= 15

GOING OUT TO Thing1 now . . .

THING1 Context:
 a= 3   b= 6
Global Context: within Thing1 (!) as called from main (!)
 a= 10  b= 20

BACK IN Main Context:
 a= 5   b= 15
```



```
//MORE SCOPE Experimenting
#include<iostream>
using namespace std;
int a = 10 , b = 20 ;
void Thing1 (int &, int );
int main()
{int a = 5 , b = 15 ;
 cout<<"\nGlobal Context: ";
 cout<< "\n a= "<< ::a << "\tb= "<< ::b;
 cout<<"\n\nMain Context: ";
 cout<<"\n a= "<< a << "\tb= "<<b;
 cout<<"\n\nGOING OUT TO Thing1 now . . .";
 Thing1(a,b);
 cout<<"\n\nBACK IN Main Context: ";
 cout<<"\n a= "<< a << "\tb= "<<b;
 cout<<"\n\n";
 return 0;
}
void Thing1 (int & a  , int b )
     {a=3;
      b=6;
      cout<<"\n\nTHING1 Context: ";
      cout<<"\n a= "<< a << "\tb= "<<b;
      cout<<"\nGlobal Context: "
          <<"within Thing1 (!) as called from main (!)";
      cout<< "\n a= "<< ::a << "\tb= "<< ::b;
     }
```

```
Global Context:
 a= 10   b= 20

Main Context:
 a= 5    b= 15

GOING OUT TO Thing1 now . . .

THING1 Context:
 a= 3    b= 6
Global Context: within Thing1 (!) as called from main (!
 a= 10   b= 20

BACK IN Main Context:
 a= 3    b= 15
```