```cpp
#include <iostream>
void Print(int *ptr) {
    using namespace std;
    cout << *ptr << endl;
}
void UsingConst() {
    using namespace std;
    float radius = 0;
    cin >> radius;
    const float PI = 3.14159f;
    float area = PI * radius * radius;
    float circumference = PI * 2 * radius;
    cout << "Area is : " << area << endl;
    cout << "Circumference is : " <<
      circumference << endl;

    ///  Pointers to identifiers with
      const qualifier
    const int CHUNK_SIZE = 512;

    int* ptr = &CHUNK_SIZE; // Not
      allowed! This is an error!

    // Initialising a pointer to a
      constant integer
    const int* ptr = &CHUNK_SIZE;
    // This means that the pointer is not
      constant,
    // but the value at the address is
      constant.
    // So reassignment of pointer works:
    int y = 10;
```

```
27      ptr = &y;  // allowed, since pointer
 •        is not const.
28      // But you are NOT ALLOWED to
 •        indirectly reassign the value at y.
29      *ptr = 5; // Error Code!!
30
31      // Initalising a CONSTANT pointer to a
 •        CONSTANT integer
32      const int *const ptr = &CHUNK_SIZE;
33      *ptr = 1; // ERROR CODE!
34      int x = 10; ptr = &x; // ERROR CODE!
35      Print(&x);
36      cout << "main->x" << x << endl;
37  }
38
39  void PrintRef(const int &ref) {
40      using namespace std;
41      cout << ref << endl;
42  }
43  void UsingConstRef() {
44      int x = 5;
45      PrintRef(1);
46  }
47  int main() {
48      UsingConst();
49      UsingConstRef();
50
51      return 0;
52  }
53
```