Step-by-step walkthrough for example on page 13 of the lecture notes:

Object Initialization,

Construction and Destruction

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
```

'c'

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
temp
```

```
#include <iostream>
                              /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
 private:
    int frequency; char* str;
 public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1];
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
temp
                                                                        'c'
                           S
                                                                            '\0'
                                       tmp
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                         'c'
                                                          'a'
                                                               'n'
                                                                             '\0'
temp
                                                         'a'
                                                                         'c'
                                                                             '\0'
                                       tmp
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
temp
                                                         'a'
                                                              'n'
                                       tmp
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                                Output:
using namespace std;
                                call implicit const char* conversion
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = ' \cdot 0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                         'c'
                                                     't'
                                                          'a'
                                                               'n'
                                                                              '\0'
temp
                                                          'a'
                                                               'n'
                                                                         'c'
                                                                              '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
 private:
    int frequency; char* str;
 public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1];
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print_word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
temp
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                                Output:
using namespace std;
                                call implicit const char* conversion
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
                Default copy constructor is called. It does memberwise copy.
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                'i'
                                                     't'
                                                          'a'
                                                               'n'
                                                                    'i'
                                                                         'c'
                                                                              '\0'
temp
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
                temp is destroyed after the expression has been evaluated!
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                         'a'
                                                                   'i'
                                                                         'c'
                                                                             '\0'
                                                              'n'
temp
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = ' \cdot 0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                               'i'
                                                    't'
                                                         'a'
                                                              'n'
                                                                   'i'
                                                                        'c'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
 private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                               'i'
                                                    't'
                                                         'a'
                                                              'n'
                                                                   'i'
                                                                        'c'
                                                                             '\0'
                                       this
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
 private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
                                                      Memory leak
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                               'i'
                                                    't'
                                                         'a'
                                                              'n'
                                                                   'i'
                                                                        'c'
                                                                             '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
    'A'
                                                      Memory leak
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
 private:
    int frequency; char* str;
 public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
            temp
                                                      Memory leak
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic: 1
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                         'a'
                                                                        'c'
                                                                             '\0'
                                                              'n'
            temp
                                                      Memory leak
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
 private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
    'A'
            temp
                                                      Memory leak
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                              'n'
                                                                             '\0'
    'A'
            temp
                                                       Memory leak
                                               'A'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
 private:
    int frequency; char* str;
 public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print_word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
            temp
                                                      Memory leak
                                                   '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic: 1
  private:
                               call implicit char conversion
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print_word('A'); return 0; }
                                                                        'c'
                                                                             '\0'
                                                              'n'
            temp
                                                       Memory leak
                                               'A'
                                                    '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic: 1
 private:
                               call implicit char conversion
    int frequency: char* str:
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
                                                      Memory leak
                                                    '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                                Output:
using namespace std;
                                call implicit const char* conversion
class Word {
                               Titanic : 1
  private:
                                call implicit char conversion
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
                Default copy constructor is called. It does memberwise copy.
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                         'c'
                                                                             '\0'
                                                               'n'
            temp
                                                       Memory leak
                                                'A'
                                                    '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
 private:
                               call implicit char conversion
    int frequency; char* str;
 public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
                temp is destroyed after the expression has been evaluated!
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                            '\0'
            temp
                                                      Memory leak
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic: 1
  private:
                               call implicit char conversion
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                       'c'
                                                                            '\0'
                                                              'n'
                                                      Memory leak
                                                   '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
                                                                                Х
class Word {
                               Titanic: 1
  private:
                               call implicit char conversion
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
                                                      Memory leak
                                                    '\0'
```

```
#include <iostream>
                               /* File: implicit-conversion-surprise.cpp */
#include <cstring>
                               Output:
using namespace std;
                               call implicit const char* conversion
class Word {
                               Titanic : 1
  private:
                               call implicit char conversion
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print word("Titanic"); print word('A'); return 0; }
                                                                        'c'
                                                                            '\0'
                                                              'n'
                                                      Memory leak
                                                    '\0'
```

```
/* File: implicit-conversion-surprise.cpp */
#include <iostream>
#include <cstring>
                              Output:
using namespace std;
                              call implicit const char* conversion
class Word {
                              Titanic : 1
  private:
                              call implicit char conversion
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1; str = new char[2]; str[0] = c; str[1] = '\0';
      cout << "call implicit char conversion\n";</pre>
    Word(const char* s) {
      frequency = 1; str = new char [strlen(s)+1]; strcpy(str, s);
      cout << "call implicit const char* conversion\n";</pre>
    void print() const { cout << str << " : " << frequency << endl; }</pre>
};
void print word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
                                                        'a'
                                                             'n'
                                                                       'c'
                                                                           '\0'
                                                      Memory leak
                                              'A' '\0' Memory leak
```