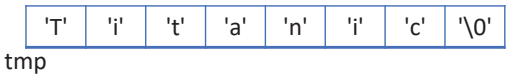
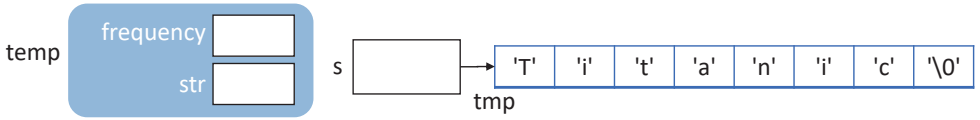
Step-by-step walkthrough for example on page 13 of the lecture notes: Object Initialization, Construction and Destruction

```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

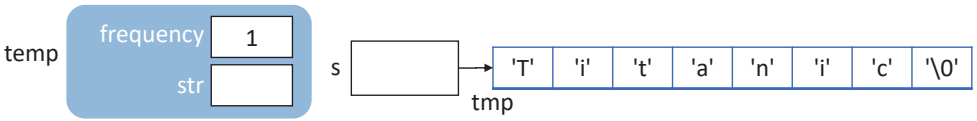
```
#include <iostream>          /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```



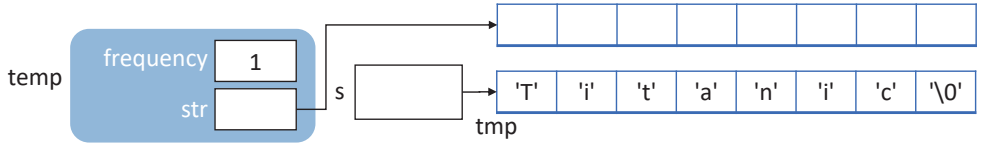

```
#include <iostream>          /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```



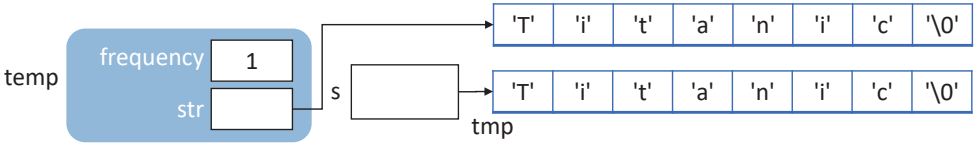
```
#include <iostream>          /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```




```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

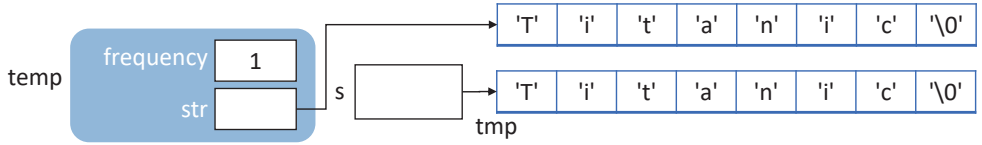



```
#include <iostream>          /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```



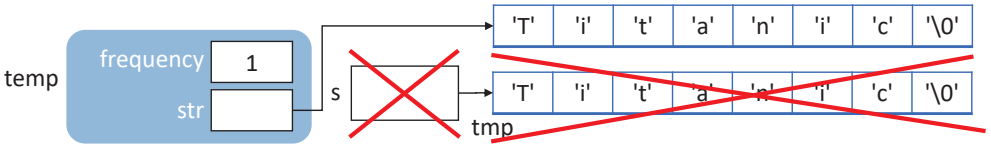

```
#include <iostream>          /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

Output:
```
call implicit const char* conversion
```



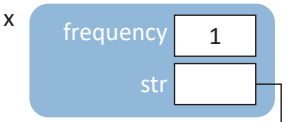
```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */

Output:
call implicit const char* conversion
```

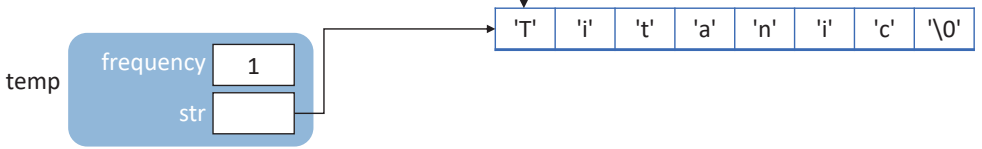



```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {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

```
/* File: implicit-conversion-surprise.cpp */

Output:
call implicit const char* conversion
```

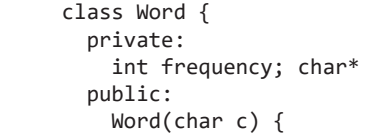


Default copy constructor is called. It does memberwise copy.

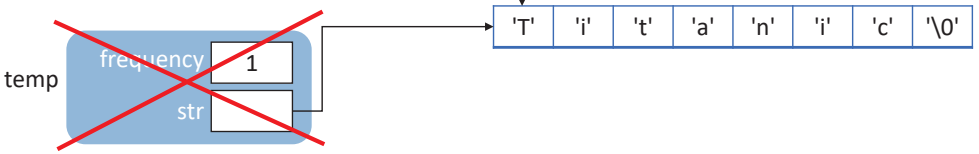



```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */

Output:
call implicit const char* conversion
```

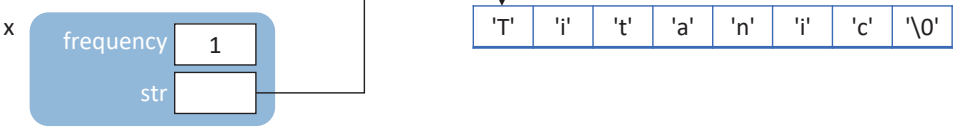


temp is destroyed after the expression has been evaluated!



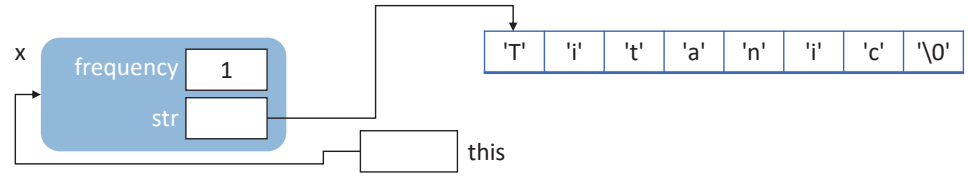

```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */

Output:
call implicit const char* conversion
```



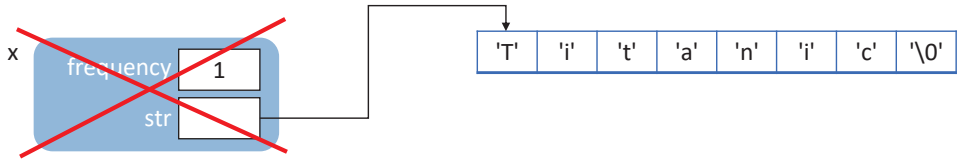
```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
```



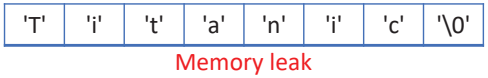

```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
```


```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
```

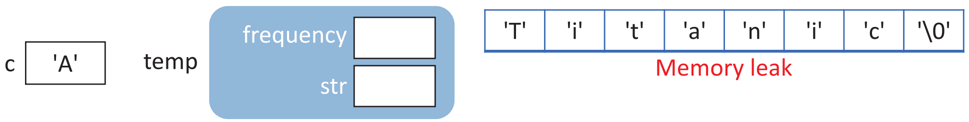




```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
```

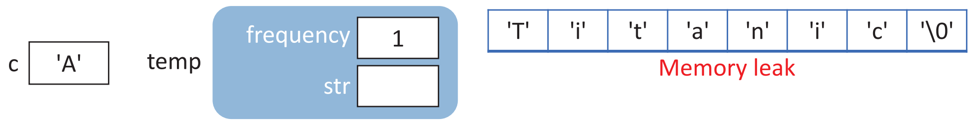

```
#include <iostream>            /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
Output:
call implicit const char* conversion
Titanic : 1
```



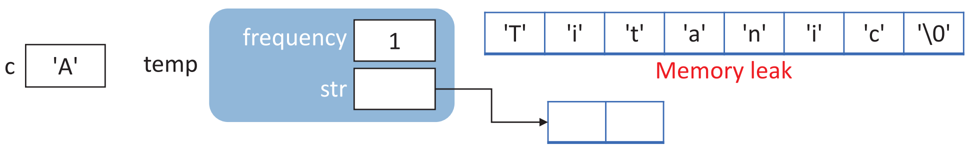

```
#include <iostream>            /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
Output:
call implicit const char* conversion
Titanic : 1
```



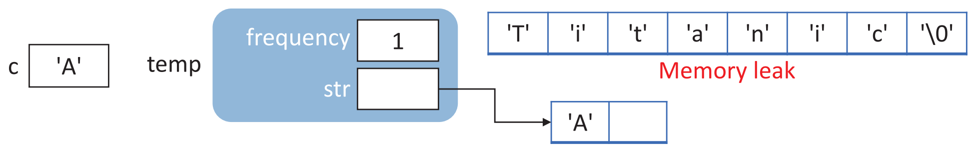

```
#include <iostream>            /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
Output:
call implicit const char* conversion
Titanic : 1
```



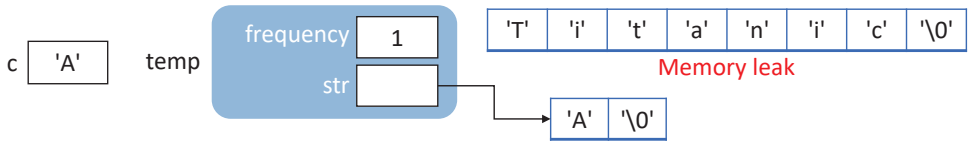

```
#include <iostream>            /* File: implicit-conversion-surprise.cpp */
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit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

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
Output:
call implicit const char* conversion
Titanic : 1
```



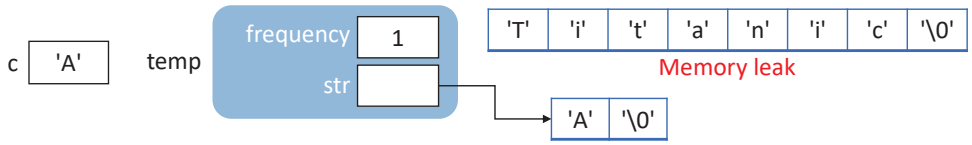
```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
```



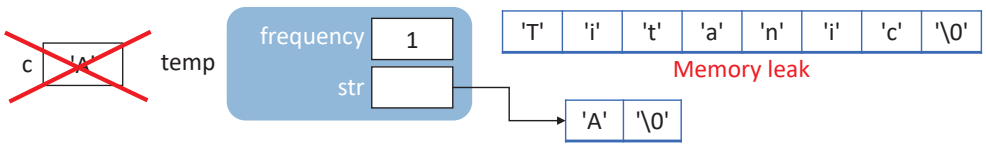

```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
```




```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
```

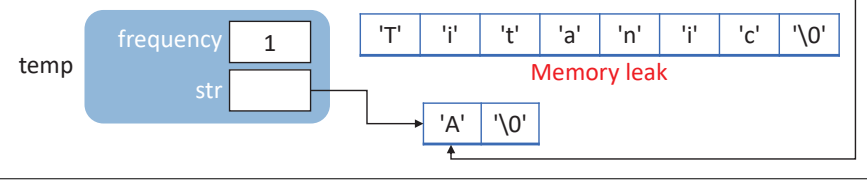



```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};
```

Default copy constructor is called. It does memberwise copy.

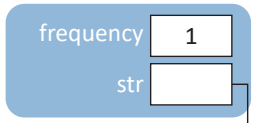
```
void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
```

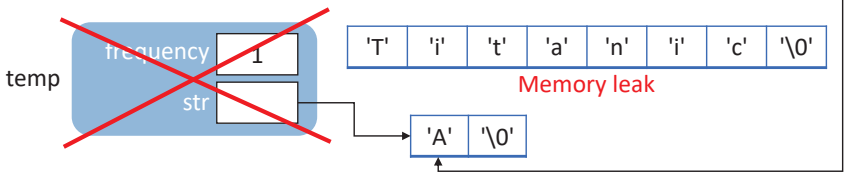


```
#include <iostream>
#include <cstring>
using namespace std;
class Word {
  private:
    int frequency; char* str;
  public:
    Word(char c) {
      frequency = 1;   str = new char[2];   str[0] = c;   str[1] = '\0';
      cout << "call implicit char conversion\n";
    }
    Word(const char* s) {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

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
```

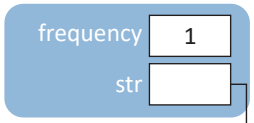


temp is destroyed after the expression has been evaluated!




```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};
void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

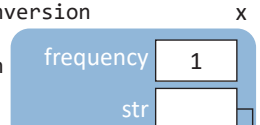
```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
```

'T' 'i' 't' 'a' 'n' 'i' 'c' '\0'
Memory leak
'A' '\0'


```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

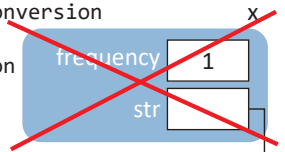
```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
A 1
```

'T' 'i' 't' 'a' 'n' 'i' 'c' '\0'
Memory leak
'A' '\0'


```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */
Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
A 1
```

'T' 'i' 't' 'a' 'n' 'i' 'c' '\0'
Memory leak
'A' '\0'

```
#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";
    }
    Word(const char* s) {
      frequency = 1;   str = new char [strlen(s)+1];   strcpy(str, s);
      cout << "call implicit const char* conversion\n";
    }
    void print() const { cout << str << " : " << frequency << endl; }
};

void print_word(Word x) { x.print(); }
int main() { print_word("Titanic"); print_word('A'); return 0; }
```

```
/* File: implicit-conversion-surprise.cpp */

Output:
call implicit const char* conversion
Titanic : 1
call implicit char conversion
A 1
```