

EXHIBIT C

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS**

CONNECTU, INC., CAMERON
WINKLEVOSS, TYLER WINKLEVOSS,
AND DIVYA NARENDRA,

Plaintiffs,

v.

FACEBOOK, INC., MARK
ZUCKERBERG, EDUARDO SAVERIN,
DUSTIN MOSKOVITZ, ANDREW
MCCOLLUM, AND THEFACEBOOK LLC,

Defendants.

)
)
) CIVIL ACTION NO. 1:07-CV-
) 10593-DPW

)
) **JURY TRIAL DEMANDED**
)
)

**EXHIBIT 37 TO THE DECLARATION OF MEREDITH H. SCHOENFELD
IN SUPPORT OF FIRST AMENDED COMPLAINT**

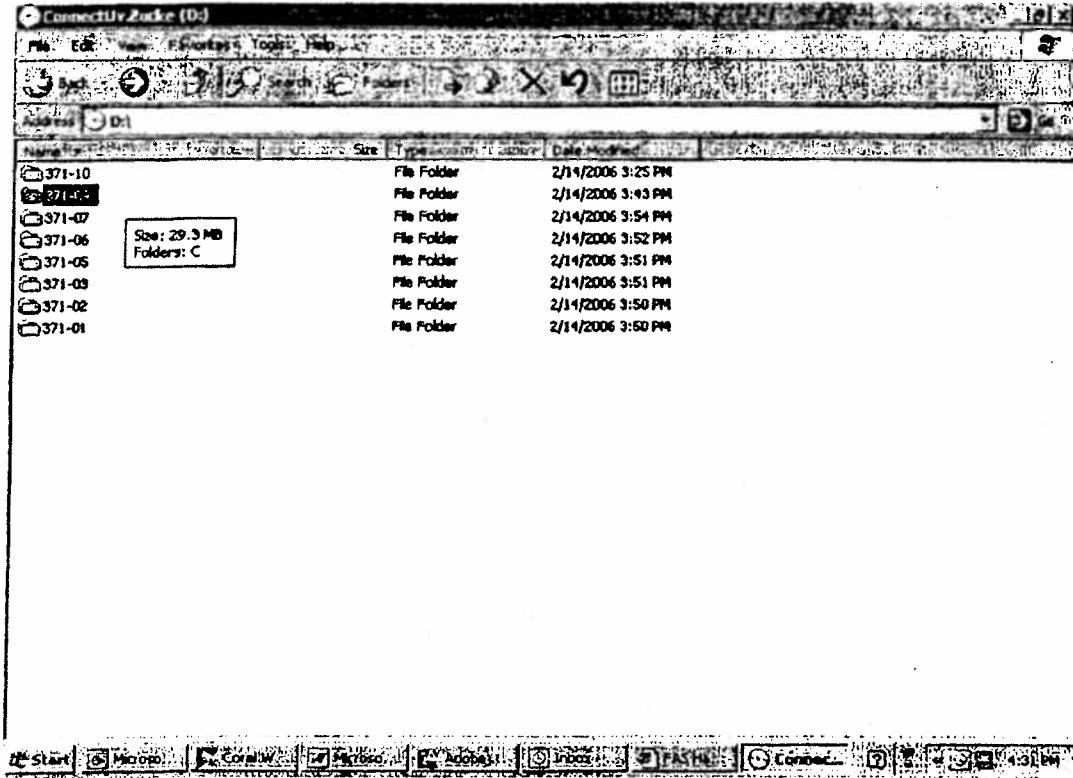
**FILED UNDER SEAL PURSUANT TO
PARTIES' STIPULATED PROTECTIVE ORDER**

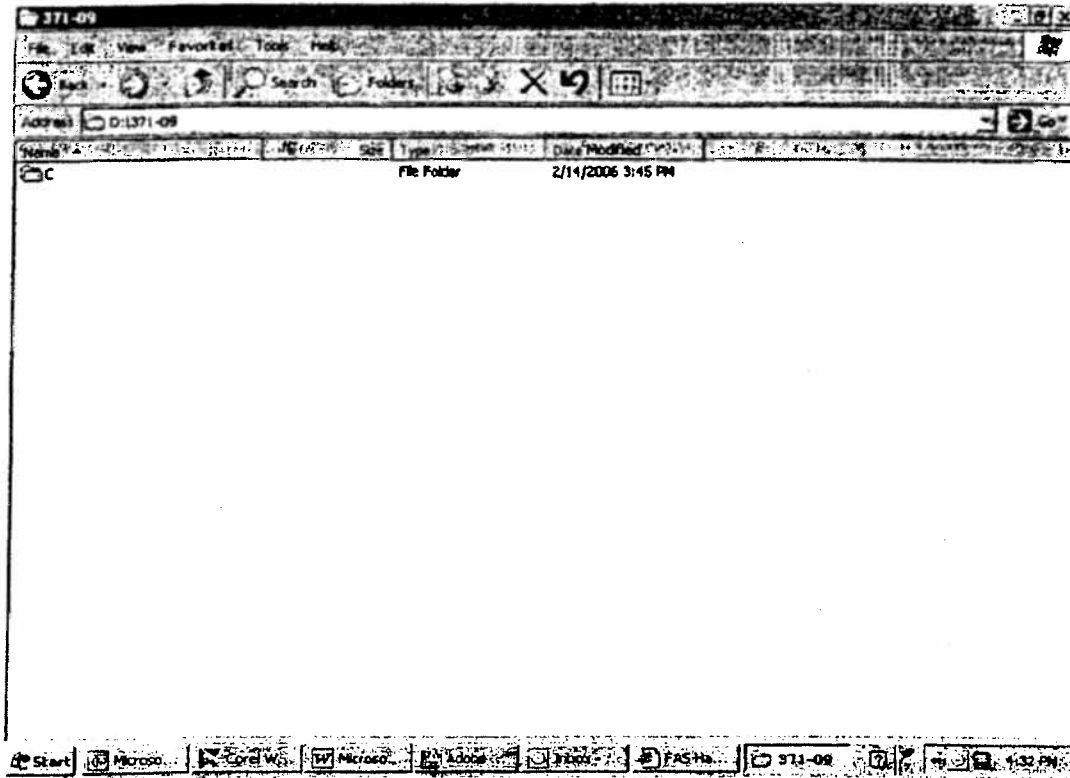
REDACTED VERSION

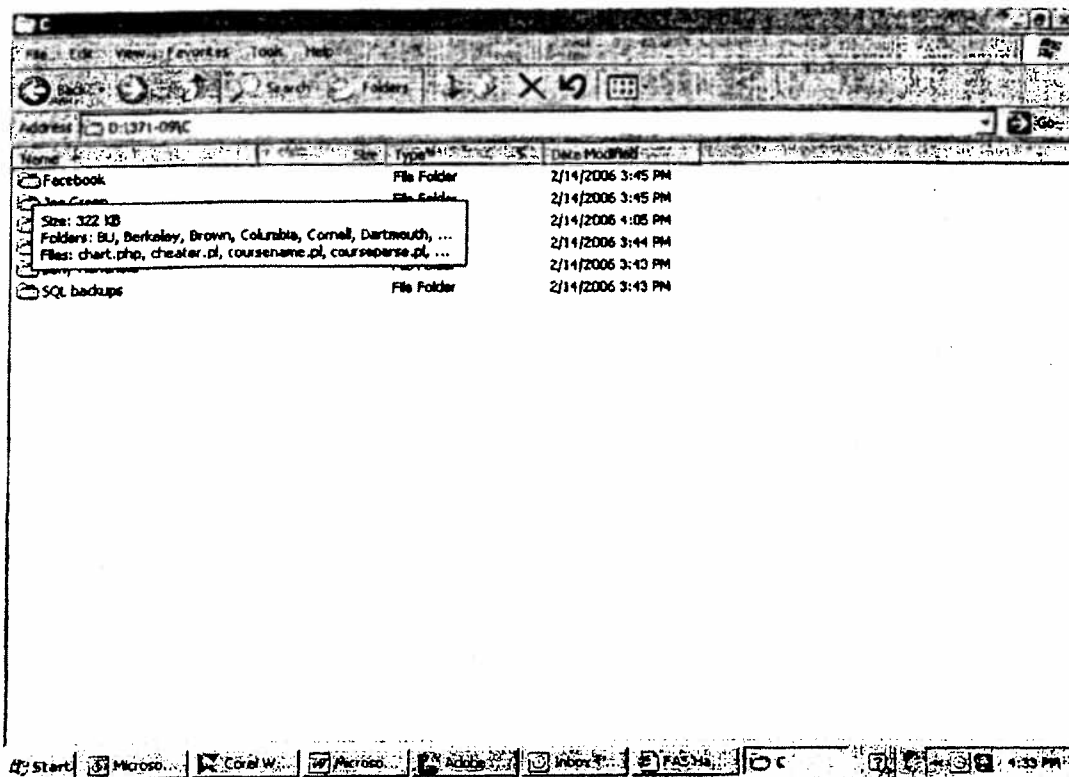
September 21, 2007

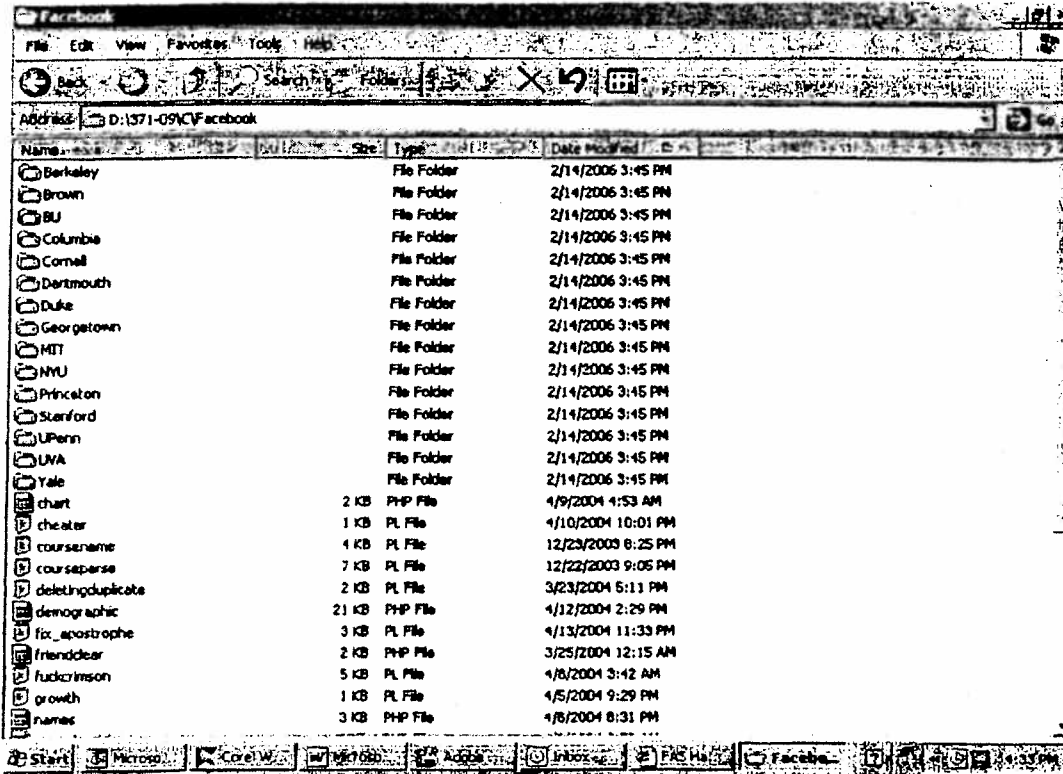
EXHIBIT 37

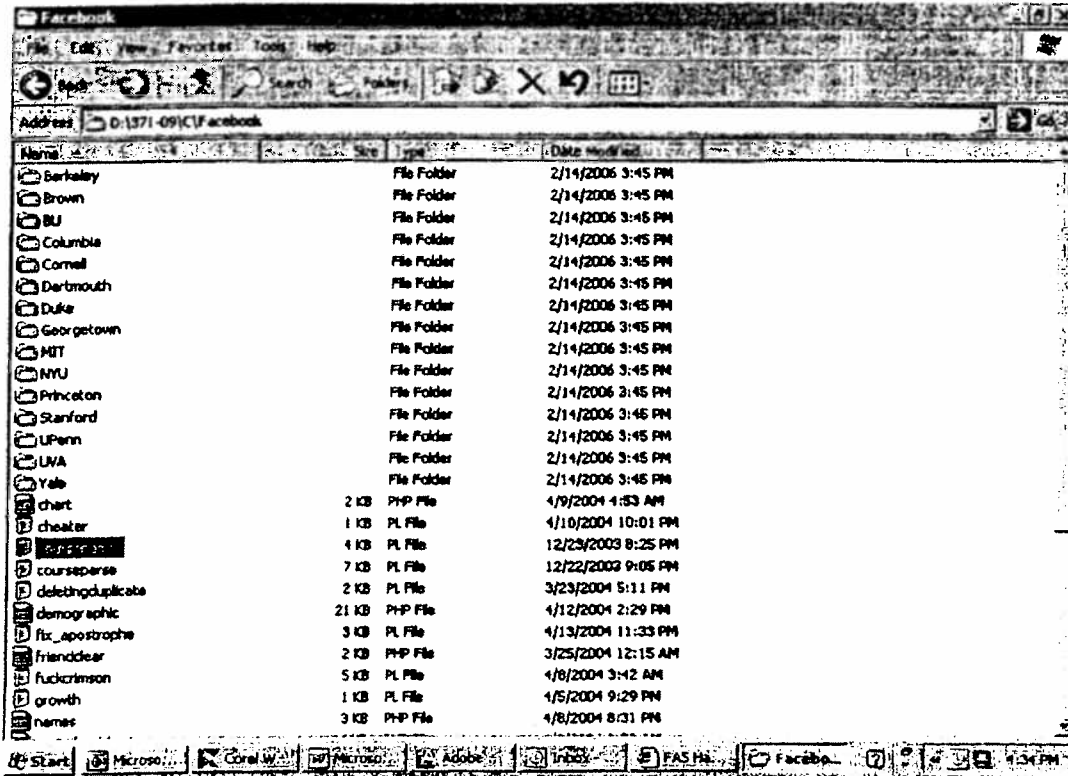


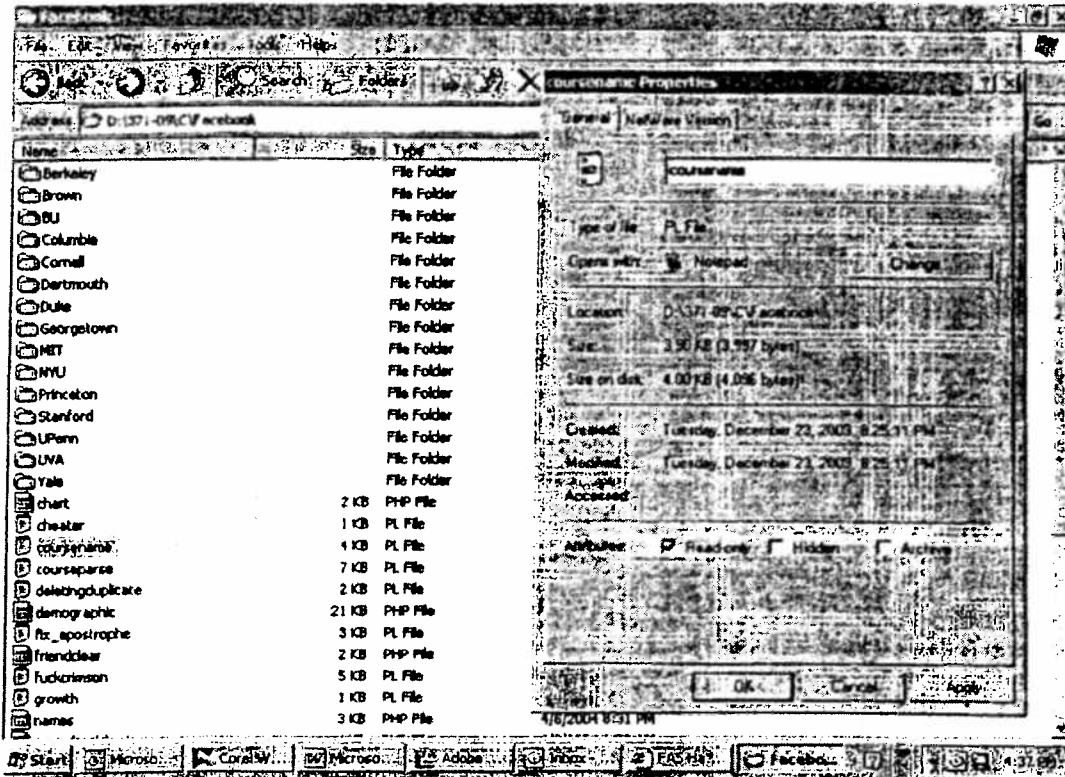












```

                                course_name

#!/usr/bin/perl

use strict;
use MySQL;

my $db = MySQL->connect("localhost", "harvardchild",
                        );
my $retval;
my $term = 0;

while (1) {
    print "> ";
    my $line = <STDIN>;
    $line =~ s/\n//g;
    find_course ($line);
}

sub find_course {
    my ($str) = @_;
    my $sql;
    my $ret;
    #my $term_clause = "(term = $term OR term = 2 OR term = 3)";
    my $term_clause = 1;
    my $order_clause = "ORDER BY ord";
    $str = lc($str);

    # see if it matches a course name directly
    $sql = "SELECT *, instr(lower(name), \"$str\") as ord " .
        "FROM course_list WHERE " .
        "instr(lower(name), \"$str\") > 0 AND $term_clause $order_clause ";
    if ($ret = fetch_course($sql)) { return $ret; }

    # see if the words make up the course name
    my @words = split / /, $str;
    $sql = "SELECT *, instr(lower(name), \"$words[0]\") as ord " .
        "FROM course_list WHERE 1";
    for (my $i = 0; $i < @words; $i++) {
        $sql .= " AND instr(lower(name), \"$words[$i]\") > 0";
    }
    $sql .= " AND $term_clause $order_clause";
    if ($ret = fetch_course($sql)) { return $ret; }

    # see if it matches a subject + number
    if ($str =~ m/^[^0-9]*([0-9]+[a-z]{0,2})/) {
        my $subject = $1;
        my $number = $2;
        while ($subject =~ m/ $/) {
            $subject =~ s/ $//;
        }
        # exceptions
        if (($subject eq "ec" or $subject eq "econ" or $subject eq "economics") and
            $number eq "10") {
            $subject = "social analysis";
        }

        my $ret_subject = $subject;
        # check for abbreviation
        for (my $i = 0; $i < 2; $i++) {
            $sql = "SELECT * FROM course_abbrev WHERE abbrev = \"$subject\"";
            $retval = $db->query($sql);
            if ($retval->num_rows) {
                my %row = $retval->fetchhash;
                $subject = lc($row{subject});
            }
        }
    }
}

```

REDACTED

```

                                course_name
@words = split / /, $subject;
$sql = "SELECT *, instr(lcase(subject), \"\$words[0]\") as ord " .
      "FROM course_list WHERE 1";
for (my $i = 0; $i < @words; $i++) {
    $sql .= " AND instr(lcase(subject), \"\$words[$i]\") > 0";
}
$sql .= " AND instr(lcase(replace(number, '-', '')), \"\$number\") > 0";
$sql .= " AND $term_clause $order_clause";
if ($ret = fetch_course($sql)) { return $ret; }
# for the 2nd time, shift chars one to the left...eg: "lac 61"->"la c61"
if ($i eq 0) {
    ($subject, $number) = left_shift($ret_subject, $number);
}
}
}

# see if the last word is really the number...eg: math xa
@words = split / /, $str;
$sql = "SELECT *, instr(lcase(subject), \"\$words[0]\") as ord " .
      "FROM course_list WHERE 1";
for (my $i = 0; $i < @words - 1; $i++) {
    $sql .= " AND instr(lcase(subject), \"\$words[$i]\") > 0";
}
$sql .= " AND instr(lcase(replace(number, '-', '')), \"\$words[@words-1]\") > 0";
$sql .= " AND $term_clause $order_clause";
if ($ret = fetch_course($sql)) { return $ret; }

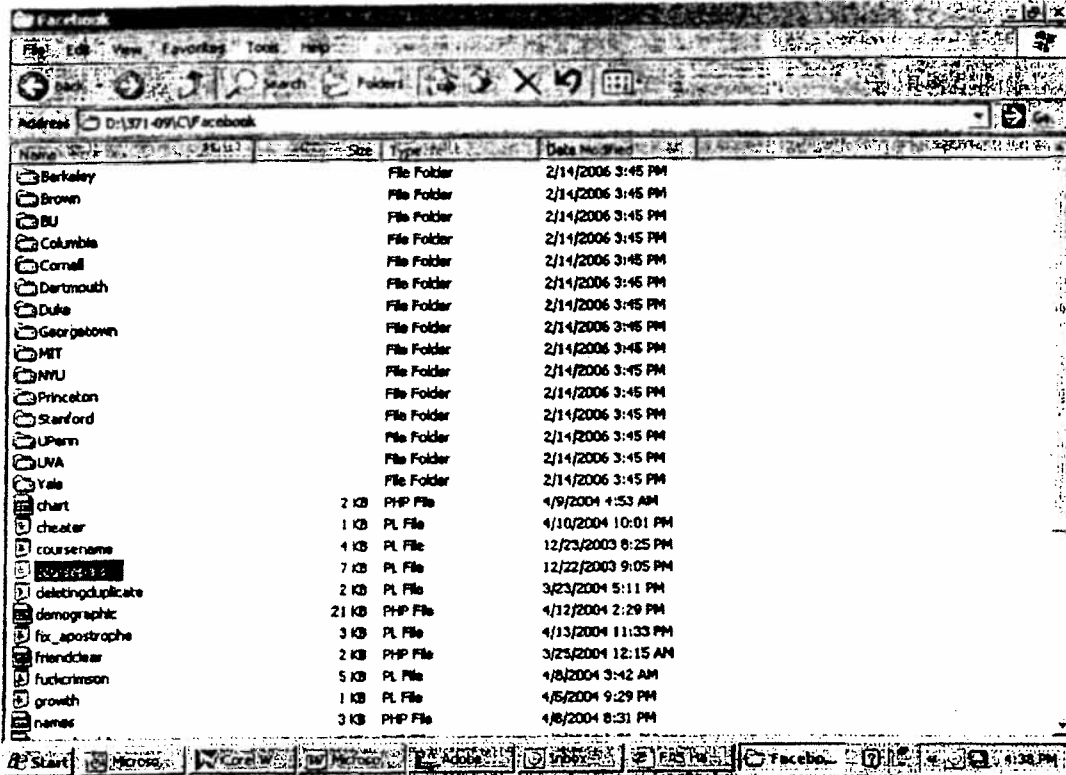
# a little exclusion in case they misspelled something
if (@words > 1) {
    for (my $i = 0; $i < @words; $i++) {
        my $first = 0;
        if ($i eq 0) { $first = 1; }
        $sql = "SELECT *, instr(lcase(name), \"\$words[$first]\") as ord " .
              "FROM course_list WHERE 1";
        for (my $j = 0; $j < @words; $j++) {
            if ($j ne $i) {
                $sql .= " AND instr(lcase(name), \"\$words[$j]\") > 0";
            }
        }
        $sql .= " AND $term_clause $order_clause";
        if ($ret = fetch_course($sql)) { return $ret; }
    }
}

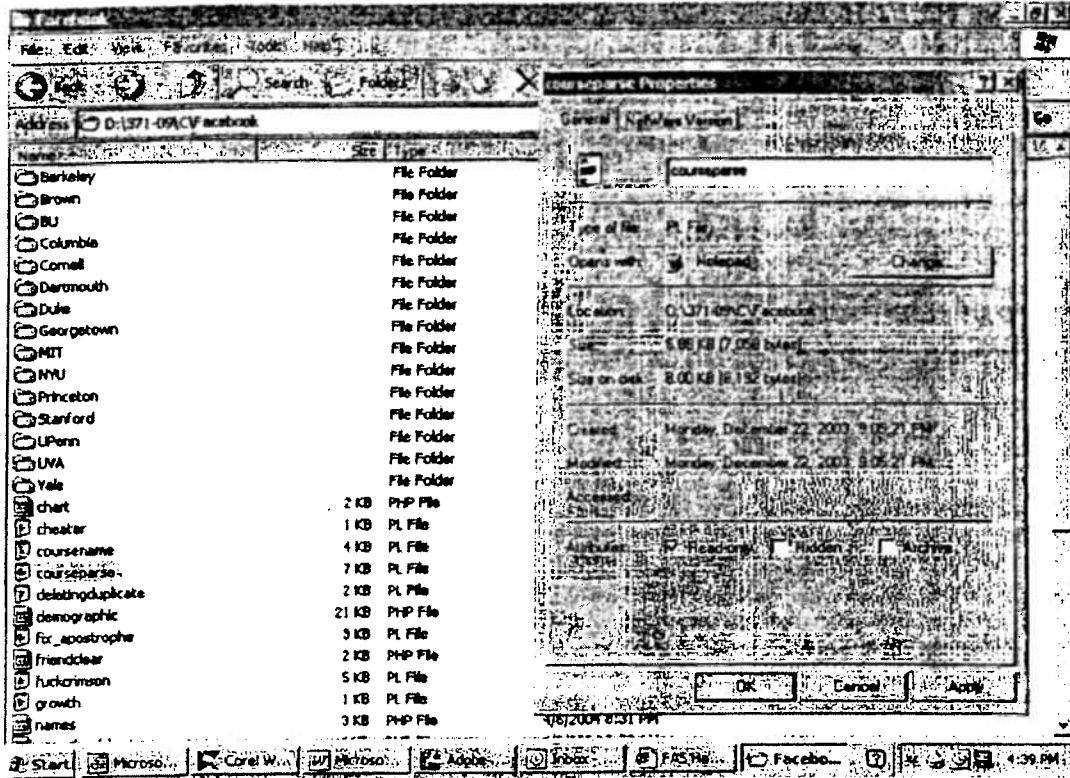
sub fetch_course {
    my ($sql) = @_;
    my $retval = $db->query($sql);
    if ($retval->num_rows eq 1) {
        my %row = $retval->fetchhash;
        print "$row{subject} $row{number}. $row{name}\n";
        return $row{id};
    } elsif ($retval->num_rows > 1) {
        my %row = $retval->fetchhash;
        #if ($row{ord} eq 1) {
            print "*$row{subject} $row{number}, $row{name}\n";
            return $row{id};
        #}
    }
    return 0;
}

sub left_shift {

```

```
                                courseName
my ($first, $second) = @_;
my @char = split //, $first;
$first = "";
for (my $i = 0; $i < @char - 1; $i++) {
    $first .= $char[$i];
}
$second = $char[@char-1] . $second;
return ($first, $second);
}
```





courseparse

```

#!/usr/bin/perl

use strict;
use Mysql;

my @course;
my @subject;
my @number;
my @name;
my @link;
my @catalog;
my @prof;
my @term;
my @days1;
my @days2;
my @start1;
my @start2;
my @end1;
my @end2;
my @exam;
my @desc;
my @note;
my $current = -1;
my $seeking = 0;
my $found_course = 0;
my $this_field = 54;
my $num_courses = 0;
my $sql;
my $core_flag = 1;
my $db = Mysql->connect("localhost", "harvardchild", "root", "corejava");

opendir DIR, "coursebook/Courses/Core";
my @files = readdir DIR;
closedir DIR;
for (my $i = 0; $i < @files; $i++) {
    open FILE, "coursebook/Courses/Core/$files[$i]";
    $num_courses = 0;
    while (my $line = <FILE>) {
        if ($line =~ m/\<strong\>/i) {
            $current++;
            $seeking = 0;
            $found_course = 0;
        } elsif ($current ne -1) {
            if (($line =~ m/\<a href=\"([^\>]*)\" \>([^\<]*)\</a\>/i) and $seeking
eq 0) {
                $course[$current] = $2;
                $link[$current] = $1;
                $course[$current] =~ s/^\*//;
                $course[$current] =~ s/\/([^\])*\)\//g;
                $course[$current] =~ s/\&\#146\;/\//g;
                $course[$current] =~ s/\&\#147\;/\//g;
                $course[$current] =~ s/\&\#148\;/\//g;
                $course[$current] =~ s/\&\#151\;/\//g;
                if ($course[$current] =~ m/^\([0-9]*\) ([^\. ]*)\. (.*)$/) {
                    $subject[$current] = $1;
                    $number[$current] = $2;
                    $name[$current] = $3;
                    $found_course = 1;
                }
            }
            $seeking++;
        } elsif (($line =~ m/catalog number: *([0-9]*)/i) and $seeking eq 1) {
            $catalog[$current] = $1;
            $seeking++;
        }
    }
}

```



```

courseparse
} elsif ($seeking eq 3) {
    $line =~ s/\n//g;
    $prof[$current] = $line;
    $prof[$current] =~ s/\&\#146\;/\&\&\&/g;
    $seeking++;
} elsif ($seeking eq 4 and ($line =~ m/<i>/i)) {
    $seeking++;
} elsif ($seeking eq 5) {
    if ($line =~ m/half course \(((\^)+)\)/i) {
        my $inside = $1;
        if (($inside =~ m/fall/i) and ($inside =~ m/spring/i)) {
            $term[$current] = 2;
        } elsif ($inside =~ m/fall/i) {
            $term[$current] = 0;
        } elsif ($inside =~ m/spring/i) {
            $term[$current] = 1;
        }
    } elsif ($line =~ m/full course/i) {
        $term[$current] = 3;
    }
    if (($line =~ m/section i/i) or ($line =~ m/sections/i)) {
        # do nothing because there are multiple sections
    } elsif ($line =~ m/fall: (.*)\; spring: (.*)/i) {
        my $fall = $1;
        my $spring = $2;
        ($days1[$current], $start1[$current], $end1[$current]) =
        parse_day($fall);
        ($days2[$current], $start2[$current], $end2[$current]) =
        parse_day($spring);
    } else {
        ($days1[$current], $start1[$current], $end1[$current]) =
        parse_day($line);
    }
    if ($line =~ m/exam\&\#32\;group:\&\#32\;[\s]*([0-9]+)/i) {
        $exam[$current] = $1;
    }
    $seeking++;
} elsif ($seeking eq 6 and ($line =~ m/(.*)\<br>/i)) {
    $desc[$current] = $1;
    $desc[$current] =~ s/\&\#146\;/\&\&\&/g;
    $desc[$current] =~ s/\&\#147\;/\&\&\&/g;
    $desc[$current] =~ s/\&\#148\;/\&\&\&/g;
    $seeking++;
    $num_courses++;
    print "adding \"$subject[$current] $number[$current].
$name[$current]\n";
    if ($term[$current] eq 2 and
        (($days1[$current] ne $days2[$current] and $days2[$current] ne
        "")) or
        ($start1[$current] ne $start2[$current] and $start2[$current]
        ne "") or
        ($end1[$current] ne $end2[$current] and $end2[$current] ne
        ""))) {
        $sql = "INSERT INTO course_list
(id, subject, number, name, catalog, link, prof, "
"term, days, start, end, exam, desc, note, year, field, core) VALUES "
"(',$subject[$current]','$number[$current]',
'$name[$current]', "
"'$catalog[$current]','$link[$current]','$prof[$current]',
'0', "
"'$days1[$current]','$start1[$current]','$end1[$current]',
'$exam[$current]', "

```

```

                                courseparse
                                "$desc[$current]", '$note[$current]', '2003',
'$this_field', '$score_flag');"
                                $db->query($sql);
                                $sql = "INSERT INTO course_list
(id, subject, number, name, catalog, link, prof, "
"term, days, start, end, exam, desc, note, year, field, core) VALUES "
"(', '$subject[$current]', '$number[$current]',
'$name[$current]', "
"$catalog[$current]', '$link[$current]', '$prof[$current]',
'1', "
"$days2[$current]', '$start2[$current]', '$end2[$current]',
'$exam[$current]', "
"$desc[$current]', '$note[$current]', '2003',
'$this_field', '$score_flag');"
                                $db->query($sql);
                                } else {
                                $sql = "INSERT INTO course_list
(id, subject, number, name, catalog, link, prof, "
"term, days, start, end, exam, desc, note, year, field, core) VALUES "
"(', '$subject[$current]', '$number[$current]',
'$name[$current]', "
"$catalog[$current]', '$link[$current]', '$prof[$current]',
'$term[$current]', "
"$days1[$current]', '$start1[$current]', '$end1[$current]',
'$exam[$current]', "
"$desc[$current]', '$note[$current]', '2003', $this_field,
score_flag)";
                                $db->query($sql);
                                }
                                #print "$sql\n";
                                } elseif ($seeking eq 7 and ($line =~ m/\<i>note: \</i>>(.*?)\<br>/i))
{
                                $note[$current] = $1;
                                $note[$current] =~ s/\&\#146\;/\&\#147\;/g;
                                $note[$current] =~ s/\&\#147\;/\&\#148\;/g;
                                $seeking++;
                                $sql = "UPDATE course_list SET note = '$note[$current]' WHERE
catalog = $catalog[$current]";
                                $db->query($sql);
                                #print "updating \"$subject[$current] $number[$current]\"...\n";
                                } elseif ($found_course eq 1 and $catalog[$current] and ($line =~
m/\<i>/i)) {
                                $seeking++;
                                }
                                }
                                }
                                close FILE;
                                if ($num_courses > 0) {
                                $sql = "INSERT INTO course_field (id, field, link) VALUES "
                                "('$this_field', '1',
'http://www.registrar.fas.harvard.edu/Courses/Core/$files[$i]')";
                                $db->query($sql);
                                $this_field++;
                                }
                                }

sub parse_day {
my ($str) = @_;
my @week = (0, 0, 0, 0, 0);
my $retval = "";

```

```

courseparse
if ($str =~ m/M\./) { $week[0] = 1; }
if ($str =~ m/Tu\./) { $week[1] = 1; }
if ($str =~ m/W\./) { $week[2] = 1; }
if ($str =~ m/Th\./) { $week[3] = 1; }
if ($str =~ m/F\./) { $week[4] = 1; }
if ($str =~ m/through/i) {
    my $flag = 0;
    for (my $i = 0; $i < 5; $i++) {
        if ($flag eq 1) {
            $week[$i] = 1;
        } elsif ($week[$i] eq 1) {
            $flag = 1;
        }
    }
}
for (my $i = 0; $i < 5; $i++) {
    if ($week[$i] eq 1) {
        $retval .= "-" . ($i+1);
    }
}
my $start;
my $end;
if ($str =~ m/(1?[0-9]):([0-9]{2})\&\#150\;(1?[0-9]):([0-9]{2})/) {
    $start = ($1*100) + $2;
    $end = ($3*100) + $4;
} elsif ($str =~ m/(1?[0-9]):([0-9]{2})\&\#150\;(1?[0-9])/) {
    $start = ($1*100) + $2;
    $end = ($3*100);
} elsif ($str =~ m/(1?[0-9])\&\#150\;(1?[0-9]):([0-9]{2})/) {
    $start = ($1*100);
    $end = ($2*100) + $3;
} elsif ($str =~ m/(1?[0-9])\&\#150\;(1?[0-9])/) {
    $start = ($1*100);
    $end = ($2*100);
} elsif ($str =~ m/(1?[0-9]):([0-9]{2})/) {
    $start = ($1*100) + $2;
    $end = (($1+1)*100) + $2;
} elsif ($str =~ m/(1?[0-9])/) {
    $start = ($1*100);
    $end = (($1+1)*100);
}
return ($retval, normalize_time($start), normalize_time($end));
}

sub normalize_time {
    my ($time) = @_;
    if ($time < 800) {
        return $time + 1200;
    } else {
        return $time;
    }
}

print "\ndone.\n";

```