

CSE 305 Programming Languages Spring, 2010

Homework 8

Maximum Points: 28

Due 10:30 AM, Friday, March 26, 2010

Professor Shapiro

March 19, 2010

Write the answers in a file named `hw8.txt`. Put your name and user name at the top of the file, and submit the file, using the UNIX command, `submit_cse305 hw8.txt`. You will be directed to submit 5 more files below.

1. (5) Consider the following C program, which you may copy from `/projects/shapiro/CSE305/ProgramsForHomeworks/goUser.c`:

```
#include <stdio.h>

int main()
{
    int i,j,k,i3,j3,i00,j0;
    i = 1;
one: if (i>9) goto six;
    i3 = i*i*i;
    i00 = 100*i;
    j = 0;
two: if (j>9) goto five;
    j3 = j*j*j;
    j0 = 10*j;
    k = 0;
three: if (k>9) goto four;
    if (i3+j3+k*k*k == i00+j0+k) goto six;
    k++;
    goto three;
four: j++;
    goto two;
five: i++;
    goto one;
six: printf("I found %d\n", i00+j0+k);
    return 0;
}
```

- (a) (2) When you run this program, what does it say that it found?
- (b) (3) Translate `goUser.c` into Java. Since Java does not have a `goto` statement, use iterative loops and at most one `break` statement. Your program must not use the `return` statement from within its loops. Call this program `NoGo.java`. Place it here in your submission file, and submit it.

2. (10) Using a case (switch) statement, write a function that implements the code of Chapter 8, Programming Exercise 3 of the text (page 388). Your function is to take one integer argument, as indicated by *k* in the text, and return the value indicated by *j* in the text. Put your function with a main program that calls the function for every integer from 1 through 8, and prints 8 lines, each showing what the function returns for each input. Name the program `multkey` with an appropriate extension
- (a) (3) Write the program in either C, C#, C++, or Java, show it here, and submit it.
 - (b) (2) Show the transcript of running the program, including the compile step.
 - (c) (3) Write the program in Ruby, show it here, and submit it.
 - (d) (2) Show the transcript of running the Ruby program.
3. (10) In lecture, I said that changing the value of a variable that is used in any of the loop parameters within the body of the loop will not affect the number of times Fortran's `Do` loop is executed, but will change the number of times the `for` loop of any of the C-based languages is executed. Here is a Fortran program containing a `Do` loop that just prints the integers from 1 to 10:

```
Program printloop
Integer i, m
m = 10
Do i = 1, m
    Print *, "i = ", i
EndDo
End
```

Copy this program from

`/projects/shapiro/CSE305/ProgramsForHomeworks/printloop.f`, add one line in the body of the loop that changes the value of *m* to some integer less than 10, and run the resulting program to see if the number of times the loop is done is less than 10.

Then rewrite your modified Fortran program in either C, C#, C++, or Java, and run that.

- (a) (3) Put your modified `printloop.f` in your submission file here, and submit it.
- (b) (2) Did your modified program loop fewer times than 10 times.
- (c) (3) Put your C, C#, C++, or Java version of the modified `printloop` program here, and submit it, naming it `printloop.c`, `printloop.cs`, `printloop.cpp`, or `Printloop.java`, whichever is appropriate.
- (d) (2) Did the program you wrote in a C-based language loop fewer than 10 times?

Continued on next page.

4. (3) Run the following Python program, which you can copy from
/projects/shapiro/CSE305/ProgramsForHomeworks/forVsWhile.py

```
#!/util/bin/python

def testFor():
    for i in range(1,4):
        print i
    return i

def testWhile():
    i = 1
    while i<4:
        print i
        i += 1
    return i

i = testFor()
print "After the for loop, i = ", i
i = testWhile()
print "After the while loop, i = ", i
```

Explain why testFor and testWhile return different values.