Which of the following does a correct swap for a and b?
(a) temp = a; b = temp; a = b;
(b) temp = a; b = a; b = temp;
(c) temp = a; b = temp; b = a;
(d) temp = a; a = b; b = temp;  1pt
What one change should be made to fix program below?

... // whatever it takes to make program correct
int main()
{
    double x;
    assign( &x, 1.6);         // (a) change &x to x
    printf("%.3f\n", *x);     // (b) change *x to x
}
double assign(double *v, double a)   // (c) change *v to v
{
    return *v = a;            // (d) change *v to v
}
What one change should be made to fix program below?

... // whatever it takes to make program correct
int main()
{
    double *xp;
    assign( &xp, 1.6);  // (a) change &xp to xp 1 pt
    printf("%.3f\n", *xp);  // (b) change *xp to xp
}
double assign(double *v, double a)  // (c) change *v to v
{
    return *v = a;  // (d) change *v to v
}

also 1 pt for “registering your presence"
downward communication
  parameters of type int, double, char, ...

upward communication
  return value of type int, double, char, ...

bi-directional communication
  parameters of type int *, double *, char *, ...
  e.g., swap(&x, &y);
Are C declarations logical?

If @ creates a pointer and * undoes this, why not use @int x; instead of int *x;

Because

int *x; means *x is an int;
int main()
{
    int x = 2010;
    onemore(&x);
    printf(..., x);
    return 0;
}

void onemore(int *a)
{
    *a = *a + 1;
    return;
}
#include<stdio.h>
#include<math.h>
double getRadius();
int main()
{
    double radius;  //RADIUS OF CIRCLE TO BE INPUT
    double area;    //AREA OF CIRCLE TO BE CALCULATED
    radius = getRadius();
    area = M_PI * radius * radius;
    printf("The area is: %.2f\n", area);
    return 0;
}

double getRadius()
{
    double radius;  //RADIUS OF CIRCLE
    printf("Enter the value of the radius: ");
    scanf("%f", &radius);
    return radius;
}
What are the objects in the previous program?
The functions, `main` and `getRadius`. The local variables `radius`, `area` in `main`, the local variable `radius` in `getRadius`.

What is the scope of the functions?
The scope of all functions we write this semester will be global. A global scope means anywhere in the program the object can be accessed.

What is the scope of the variables?
Every variable in this program has a local scope, that is, the variables can only be referenced from within the function they have been declared.
If a common function has subfunctions, list these only the first time.
Line editing for csh

navigation:
    arrow keys, ^f, ^b, ^p, ^n, esc-f, esc-b, ^a, ^e