CS 101: Computer Programming and Utilization

Shivaram Kalyanakrishnan
(Abhiram Ranade's slides, borrowed and edited)
Lecture 12

Today's Lecture

- Call by reference
- Program demo

Some shortcomings for now

Using what we just learned, it is not possible to write functions to do the following:

- A function that exchanges the values of two variables.
- A function that produces several values as results:
  - Function to produce polar coordinates given Cartesian coordinates.

Exchanging the values of two variables, attempt 1

```
void exchange(int m,
int n){
   int temp = m;
   m = n; n = temp;
   return;
}
main_program{
   int a=1, b=2;
   exchange(a,b);
   cout << a <<‘ ‘<<
           b << endl;
}
```

- Does not work. 1, 2 will get printed.
- When `exchange` is called, 1, 2 are placed into `m, n`.
- Execution of `exchange` exchanges values of `m,n`.
- Change in `m,n` does not affect the values of `a,b` of `main_program`.

Reference parameters

```
void exchange(int &m,
int &n){
   int temp = m;
   m = n; n = temp;
   return;
}
main_program{
   int a=1, b=2;
   exchange(a,b);
   cout << a <<‘ ‘<<
           b << endl;
}
```

- & before the name of the parameter:
- Says: “Do not allocate space for this parameter, but instead just use the variable from the calling program.”
- When function changes `m,n` it is really changing `a,b`.
- Such parameters are called reference parameters.
- 2 1 will be printed.

Remark

- If a certain parameter is a reference parameter, then the corresponding argument is said to be “passed by reference”.
- Now we can write a program that computes polar coordinates given cartesian coordinates
  - We use two reference parameters.
  - Called function stores the polar coordinates in the reference parameters.
  - These changes can be seen my the main program.
- There are other ways of returning 2 values – study later.

Cartesian to polar

```
void CtoP(double x,
double y,
          double &r,
double &theta){
   r = sqrt(x*x + y*y);
   theta = atan2(y, x);
//arctan
return;
}
main_program{
   double x=1, y=1, r,
theta;
   CtoP(x,y,r,theta);
   cout << r <<‘ ‘<<
theta << endl;
}
```

- r, theta in CtoP are reference parameters,
- changing them in CtoP changes the value of r, theta in the main program.
- Hence sqrt(2) and pi/4 (45 degrees) will be printed.

What we discussed

- If we want to return more than one result we can do so by using a reference parameter.
- If we use a reference parameter `R` in a function, and pass as argument a variable `A`, then any change that the function makes in `R` will be seen by the calling program as a change in `A`.

Today's Lecture

- Call by reference

- Program demo