

## **Location Of The Online Examples**

- WWW:
  - [www.cpsc.ucalgary.ca/~tamj/231/examples/list\\_functions](http://www.cpsc.ucalgary.ca/~tamj/231/examples/list_functions)
- UNIX:
  - /home/231/examples/list\_functions

James Tam

## **Important Things To Keep In Mind**

- (What you should now): Lists are a composite type that can be decomposed into other types.
- Other important points:
  - Copying lists
  - Passing lists as parameters

James Tam

## Copying Lists

- Reminder:

- A list variable is not actually a list!
- Instead that list variable is actually a reference to the list.
- (This is important because if you use the assignment operator to copy from list to another you will end up with only one list).

```
list1 = [1,2]
list2 = [2,1]
print (list1, list2)
```

```
list1 = list2
print (list1, list2)
```

```
list1[0] = 99
print list1, list2
```

James Tam

## Copying Lists (2)

Reminder:

- To copy the elements of one list to another a loop is needed to copy each successive elements.

```
list1 = [1,2,3,4]
list2 = []
```

```
for i in range (0, 4, 1):
    list2.append(list1[i])
```

```
print (list1, list2)
list1[1] = 99
print (list1, list2)
```

James Tam

## Parameter Passing

- What you've seen so far:
  - Passing a parameter into a function makes a local copy of the value passed in.
  - This is referred to as **PASS BY VALUE**.
  - Changes made to the parameter will only be made to the local copy and not the original.

James Tam

## Parameter Passing (2)

- Passing lists into functions is done using a different mechanism
  - When a list is passed into the function a local reference refers to the original list.
  - Example:
  - Name of the online example: parameter1.py

```
def fun (list):  
    list[0] = 99  
    print (list)
```

```
def main ():  
    list = [1,2,3]  
    print (list)  
    fun (list)  
    print (list)
```

```
main ()
```

- Changes made to the local reference will change the original list.
- This parameter passing mechanism is referred to as **PASS BY REFERENCE** (the local reference refers to the original list)

James Tam

## Parameter Passing (3)

- Exception: if the local reference is assigned to another list then it will obviously no longer refer to the original list.
- (Effect: changes made via the local reference will change the local list and not the original that was passed into the function).
- Name of the online example: parameter2.py

```
def fun (list):  
    list = [3,2,1]  
    print list
```

```
def main ():  
    list = [1,2,3]  
    print list  
    fun (list)  
    print list
```

```
main ()
```