## Lecture 14.1

Topics

1. Arrays and Returns of Functions – Examples
2. Passing-by-Reference in Function – Pointers as Argument

---

## 1. Arrays and Returns of Functions – Examples

What are going on with the following functions?

**Examples – C version**

```
int* foo1( void ) { // C version
  int iSize = 3;
  int* iPtr;
  int iAry[ 4 ] = { 1, 9, 25 };

  iPtr = ( int* ) malloc( sizof( int ) * iSize );

  for ( int i = 0; i < iSize; i++ ) {
    *( iPtr + i ) = i * i;
  }

  return iPtr;
}

int* foo3( void ) { // C version
  int iSize = 3;
  int* iPtr;
  int iAry[ 4 ] = { 1, 9, 25 };

  iPtr = ( int* ) malloc( sizof( int ) * iSize );

  for ( int i = 0; i < iSize; i++ ) {
    *( iPtr + i ) = i * i;
  }

  free( iPtr );

  return iPtr;
}
```

**Examples – C++ version**

```
int* foo2( void ) { // C++ version
  int iSize = 3;
  int* iPtr;
  int iAry[ 4 ] = { 1, 9, 25 };

  iPtr = new int[ iSize ];

  for ( int i = 0; i < iSize; i++ ) {
    *( iPtr + i ) = i * i;
  }

  return iAry;
}

int* foo4( void ) { // C++ version
  int iSize = 3;
  int* iPtr;
```

```
    int iAry[ 4 ] = { 1, 9, 25 };

    iPtr = new int[ iSize ];

    for ( int i = 0; i < iSize; i++ ) {
      *( iPtr + i ) = i * i;
    }

    delete [] iAry;

    return iPtr;
}
```

**Discussion will be given in class for the above examples.**

## 2. Passing-By-Reference in Functions – Pointer as Argument

Pointers can also be sent to functions. In this brief discussion of the use of pointer as argument in function, let's look at the general prototypes and consider several simple functions as examples.

To send a pointer to a function, the function prototype needs to reflect an argument as a pointer. This is done as follows,

```
void functionName( DataType * );
```

where

- **`functionName`** is the function name
- **`DataType`** is any valid data type that is available

What does it mean regarding the use of the pointer in the above prototype?

The answer is that the function in the given prototype will receive an address of a specified type. That means an address of type **`DataType`** must be sent to the function as an argument in the above general prototype.

Let's look at several sample prototypes.

**Prototype #1**

```
void print( int* ); /* printing out the value stored at
                       a given address */
```

**Prototype #2**

```
void displayAryA( int[], int ); /* displaying the array
                                   stored at a given
                                   address and size */
```

**Prototype #3**

```
void displayAryB( double*, int ); /* displaying the array
                                     stored at a given
                                     address and size */
```

**Prototype #4**

```
void swap( int*, int* ); /* swapping two int values stored
                            at a given addresses */
```

**Discussion will be given in class for the above examples.**