PASSING ARRAYS AS FUNCTION ARGUMENTS IN OBJECTIVE-C

If you want to pass a single-dimensional array as an argument in a function, you would have to declare function formal parameter in one of following three ways and all three declaration methods produce similar results because each tells the compiler that an integer pointer is going to be received. Similar way, you can pass multi-dimensional array as formal parameters.

Way-1

Formal parameters as a pointer as follows. You will study what is pointer in next chapter.

```
- (void) myFunction(int *) param
{
.
.
.
}
```

Way-2

Formal parameters as a sized array as follows:

```
- (void) myFunction(int [10] )param
{
.
.
.
}
```

Way-3

Formal parameters as an unsized array as follows:

```
-(void) myFunction: (int []) param
{
.
.
.
}
```

Example

Now, consider the following function, which will take an array as an argument along with another argument and based on the passed arguments, it will return average of the numbers passed through the array as follows:

```
-(double) getAverage:(int []) arr andSize:(int) size
{
  int    i;
  double avg;
  double sum;

  for (i = 0; i < size; ++i)
  {
    sum += arr[i];
  }

  avg = sum / size;

  return avg;
}
```

Now, let us call the above function as follows:

```
#import <Foundation/Foundation.h>

@interface SampleClass:NSObject

/* function declaration */
-(double) getAverage:(int []) arr andSize:(int) size;

@end

@implementation SampleClass

-(double) getAverage:(int []) arr andSize:(int) size
{
  int    i;
  double avg;
  double sum =0;

  for (i = 0; i < size; ++i)
  {
    sum += arr[i];
  }

  avg = sum / size;

  return avg;
}

@end

int main ()
{
   /* an int array with 5 elements */
   int balance[5] = {1000, 2, 3, 17, 50};
   double avg;

   SampleClass *sampleClass = [[SampleClass alloc]init];
   /* pass pointer to the array as an argument */
   avg = [sampleClass getAverage:balance andSize: 5] ;

   /* output the returned value */
   NSLog( @"Average value is: %f ", avg );

   return 0;
}
```

When the above code is compiled together and executed, it produces the following result:

```
2013-09-14 03:10:33.438 demo[24548] Average value is: 214.400000
```

As you can see, the length of the array doesn't matter as far as the function is concerned because Objective-C performs no bounds checking for the formal parameters.