

# Lecture 11:

## Passing Arrays to Functions

Prof. Shervin Shirmohammadi  
University of Ottawa

# Passing Arrays as Arguments to Functions

## Passing One Dimensional Arrays to Functions

- The one dimensional array is passed to a function simply by citing the name of the array in the call to the function.
  - The array name is cited **without** `[]`.
  - The number of elements can be included in the call if the size of the array is required by the function.
  - The name of the array is a reference to the array (its address).

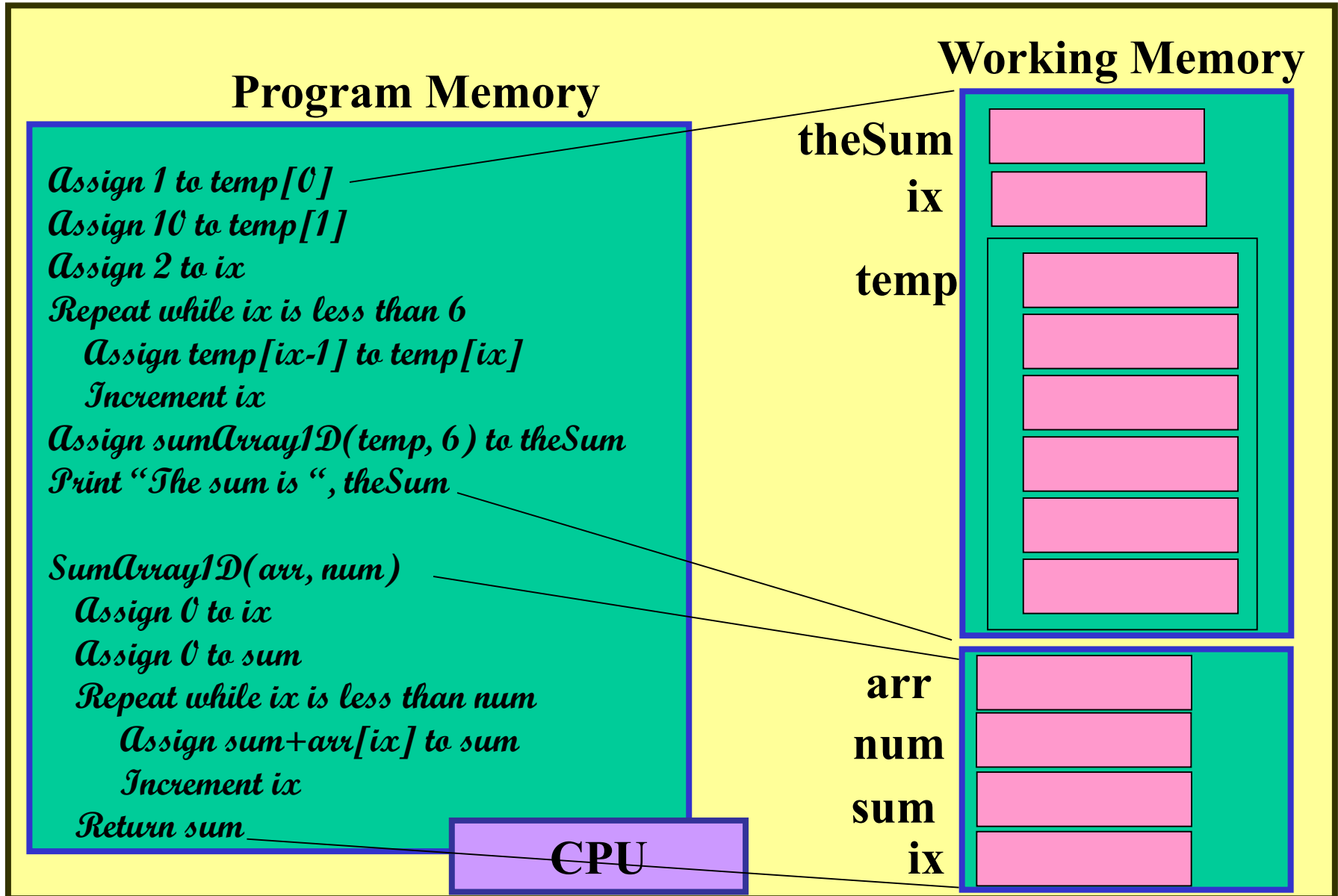
E.g.:     `int temp[5]={0};`  
          `⋮`

`sum_array_1D(temp, 5);` ← **Function call in main()**.

- The definition of a function that receives a one dimensional array must specify that an array will be received through its parameter list.
  - The passage of an array is indicated by adding `[]` after the name used to access the array in the function.
  - The brackets can be empty or may contain the number of elements in the array.

E.g.:     `void sum_array_1D(int array_1D[], int dim)`  
          or:     `void sum_array_1D(int array_1D[5], int dim)`

# Passing arrays and matrices to subprograms



# Example

- Write a function that adds up the values in an array, and test the function in a program with two different arrays.

# Passing Two Dimensional Arrays to Functions

- An entire two dimensional array is passed to a function simply by citing the name of the array.
  - The array name is cited **without** `[] []`.
  - The number of rows and columns can be included in the call if required by the function.
  - The name of the matrix is a reference to the matrix.

```
E.g.:  int a[3][4]={0};
        :
        sum_array_2D(a, 3, 4);
```

- The definition of a function that receives a two dimensional array must specify that a two dimensional array will be received through its parameter list.
  - The passage of a two dimensional array is indicated by adding `[] [4]` after the array name
  - The **column brackets** must contain the number of columns in the array.
  - The **row brackets** can be empty or may contain the number of rows in the array.

```
E.g.: void sum_array_2D(int array_2D[] [4], int rows, int cols)
      or: void sum_array_2D(int array_2D[3] [4], int rows, int cols)
```

- The prototype of the function must also specify that a two dimensional array will be received through its parameter list; this is stated in a manner similar to the function definition.

```
E.g.: void sum_array_2D(int [] [4], int, int);
      or: void sum_array_2D(int array_2D[3] [4], int rows, int cols);
```

# Example

- Write a program to calculate the total grades for each student using a function called `computeGrades()`

Student	Lab	Midterm	Final	Assignments	Total
Alex	20	23	19	22	???
Fred	19	22	22	17	???
Maria	16	21	24	19	???
John	25	14	25	22	???

# On the Passage of Array as Arguments to a Function

- In C, an array is passed by reference to a function and not by value as is done with variables.
  - An array that is modified in a function will appear modified in the caller as well.
- In C, an array element is passed by value to a function as though it were a variable.
  - The array element can be modified in the function and this modification will not appear in the caller.

# Example

- Write a program that tries to change element 2 of an array of size 5, once by passing the array and once by passing the element to two different functions, respectively.