User-defined Functions

Dr. Asif Uddin Khan

Type of User-defined Functions in C

4 types of user

1. Function with no arguments and no return value
2. Function with no arguments and a return value
3. Function with arguments and no return value
4. Function with arguments and a return value

Example: Function with no arguments and no return value

```
#include<stdio.h>
void greatNum();// function declaration
int main()
{
   greatNum(); // function call
   return 0;
}
void greatNum()// function definition
{
   int i, j;
   printf("Enter 2 numbers");
   scanf("%d%d", &i, &j);
   if(i > j) {
      printf("The greater number is: %d", i);
   }
   else {
      printf("The greater number is: %d", j);
   }
}
```

Passing arrays to a function in C

- In C programming, you can pass entire array to functions.
- Before we learn that, let's see how you can pass individual elements of an array to functions.

Passing individual array elements

- Passing array elements to a function is similar to passing variables to a function.

Example 1: Passing an array

```c
#include <stdio.h>
void display(int age1, int age2)
{
    printf("%d\n", age1);
    printf("%d\n", age2);
}

int main()
{
    int ageArray[] = {2, 8, 4, 12};

    // Passing second and third elements to display()
    display(ageArray[1], ageArray[2]);
    return 0;
}
```

Example 2: Passing entire array to functions

- To pass an entire array to a function, only the name of the array is passed as an argument.

```c
// Program to calculate the sum of array elements by passing to a function

#include <stdio.h>
float calculateSum(float age[]);

int main() {
    float result, age[] = {23.4, 55, 22.6, 3, 40.5, 18};

    // age array is passed to calculateSum()
    result = calculateSum(age);
    printf("Result = %.2f", result);
    return 0;
}

float calculateSum(float age[]) {

  float sum = 0.0;

  for (int i = 0; i < 6; ++i) {
                sum += age[i];
  }

  return sum;
}
```

Second largest array element in C

```
int array[10] = {101, 11, 3, 4, 50, 69, 7, 8, 9, 0};
int loop, largest, second;

if(array[0] > array[1]) {
   largest = array[0];
   second  = array[1];
} else {
   largest = array[1];
   second  = array[0];
}

for(loop = 2; loop < 10; loop++) {
   if( largest < array[loop] ) {
      second = largest;
      largest = array[loop];
   } else if( second < array[loop] ) {
      second =  array[loop];
   }
}
```