

Appointment 19/06/2020

Notes

### • Functions with Array Parameters:

An array cannot be passed by value to a function. However, an array name is a pointer, so just passing an array name to a function is a passing a pointer to the array.

### • Strings:

A string in C is an array of characters that ends with a Null character '\0'.

```
char str_name[str_len] = "string";
```

### • Array of strings:

A two dimensional array can be used to store related strings.

```
char trip[3][5] = {  
    "suitcase"  
    "passport"  
    "ticket"  
};
```

### • Function Pointers:

\* A function pointer declaration uses the just as you would with any point:

```
return-type (*func_name)(parameters)
```



## • Accessing Structure Members:

The members of a struct var.

sl.age = 19;

## • Using typedef.

The typedef keyword creates a type definition that simplifies code and makes a program easier to read.

```
typedef struct {  
    int id;  
    char title[40];  
    float hours;  
} course  
course cs1;  
course cs2;
```

## • Structures with structures:

The members of a structure may also be structures.

```
typedef struct {  
    int x;  
    int y;  
} point;  
typedef struct {  
    float radius;  
    point center;  
} circle;
```



## • Unions:

A union declaration uses the keyword union, a union tag, and curly braces {} with a list of members.

```
union val {
```

```
    int int-num;
```

```
    float fl-num;
```

```
    char str[20];
```

```
};
```

Ex: union val U1;

union val U2;

U2 = U1;

→ Accessing Union Members:

→ Structures with Unions:

→ Pointers to Unions.

→ Unions as Function Parameters:

→ Array of Union.

## • Memory Management.

```
int x;
```

```
printf("%d", sizeof(x)); // output: 4
```

## • Memory Management Functions.



## • The malloc Function:

The malloc() function allocates a specified number of contiguous bytes in memory.

`ptr = malloc (100 * sizeof(*ptr));`

→ The malloc Function:

→ The free function:

## • The calloc Function:

The calloc() function allocates memory based on the size of a specific item, such as a structure.

`recs = calloc (num-recs, sizeof(record));`

## • The realloc Function:

The realloc() function expands a current block to include additional memory.

`ptr = realloc(ptr, 100 * sizeof(*ptr));`

## • Allocating Memory for strings:

### • Dynamic Arrays

### • The exit Command:

• Using `errno`: