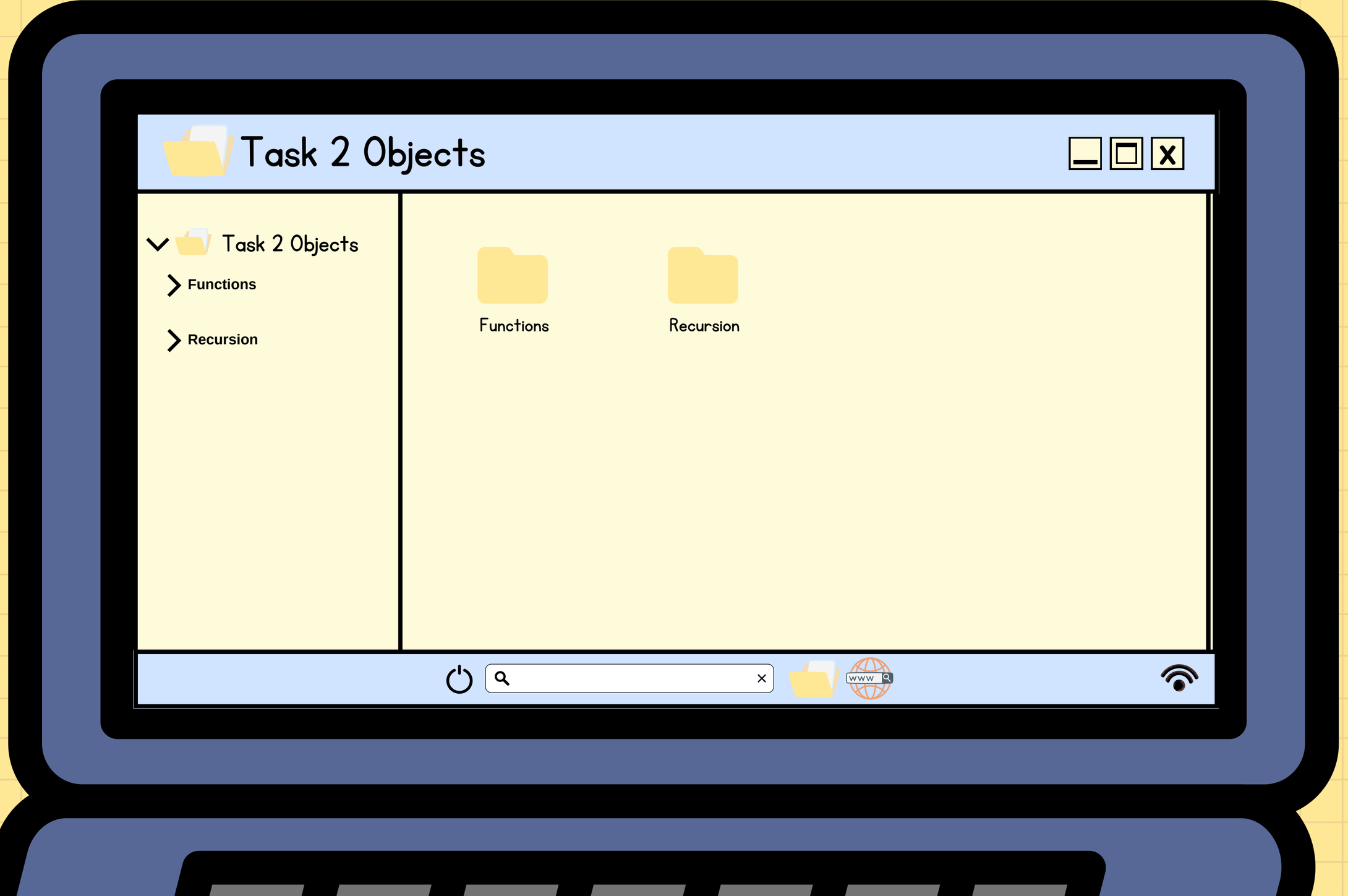
Embedded Systems

Task 2

by Mohammed Elahmady


Task 2 Objects
Task 2 Objects
Functions
Recursion
Functions
Recursion
www

Functions

Functions

- A set of statement that together to perform a specific task.
- Every C Program consists one or more functions.
- the main() function is mandatory for the c program because it is the entry point of your c code from where your program executed.

Advantages of Function :

- The function increases the modularity of the program.
- a large problems can be divided into sub programs and then solved by calling functions.
- the function increases the reusability because functions are reusable.
- the function increases the modularity of your program, so the program becomes more maintainable.

Functions

Types of functions :

1. **Library Functions :**

- like other languages, C has many built in libraries, Ex : for I/O operation printf, scanf .
- Before using any library function, you must include the corresponding header file “we must include library header file” .

2. User-Defined Functions :

- we can also create a functions according to our requirements .
- before creating your own functions you need to know about three aspects functions :

Functions

- **Function Declaration :**

```
int add(int x , int y);
```

- **Function Definition:**
  - Function Definition : contains single or group of statements that perform specific task .
  - the function definition can be categorized into two parts : Function Header , Function Body .

```
int add(int x , int y) // Function Header
{
    return (x+y);      // Function Body start from the scope {}
}
```

Functions

- **Function Call :**

```
int sum = add(24,30); // sum = 54
```

Functions

User-Defined function shapes :

1. **Void Function :**

```
void add (void);

void add (void){
  int x , y ;
  scanf("%d %d",&x,&y);
  printf("the sum = %d\n",(x+y));
}

add();
```

2. **Void Return Function + Parameters :**

```
void add (int x, int y);

void add (int x, int y){
  printf("the sum = %d\n",(x+y));
}

add(5,4);
```

Functions

User-Defined function shapes :

3. **Return Function void Parameter :**

```c
int add (void);

int add (void){
  int x , y ;
  scanf("%d %d",&x,&y);
  return (x+y);
}

int sum = add();
```

4. **Return Function + Parameter :**

```c
int add (int x, int y);

int add (int x, int y){
  return (x+y);
}

int sum = add(12,7);
```

Recursion

Recursive Functions

- A Recursive Function is a function that calls itself.

Types of Recursive Functions :

1. **Direct Recursion** : A function that calls itself within its own implementation.

```c
void print (void){
    printf("Mohammed\n");
    sleep(250);
    printf();
}
```

Recursion

2. **Direct Recursion** : A function that calls itself within its own implementation.

```c
void print_Mohammed (void){
    printf("Mohammed\n");
    sleep(250);
    printf_Ahmed();
}
```

```c
void print_Ahmed (void){
    printf("Ahmed\n");
    sleep(250);
    print_Mohammed();
}
```

```c
unsigned int factorial(unsigned int n){
  if(n < 1){
      return 1;
  }
  else{
      return(n*factorial(n-1));
  }
}
```

- The problem with recursion is that it takes up a lot of stack memory, and if the memory is exhausted, a stack overflow occurs.

THANK YOU

Head : Tasnem Sabry

Vice : Ahmed Yasser