

# Informační a komunikační technologie

## Funkce v C

**David Weber**

Kabinet K13

[weber3@spsejecna.cz](mailto:weber3@spsejecna.cz)

# Příklad na úvod

```
// Print array
for (int i = 0; i < size; i++) {
    printf("%d ", arr[i]);
}
printf("\n");
// Square array elements
for (int i = 0; i < size; i++) {
    arr[i] *= arr[i];
}
// Print array
for (int i = 0; i < size; i++) {
    printf("%d ", arr[i]);
}
```

# V čem je problém?

- Kód je funkční, ale část pro výpis pole je zde uvedena **dvakrát**.

```
for (int i = 0; i < size; i++) {  
    printf("%d ", arr[i]);  
}
```

- Opakující kód je nepraktický ✗
  - budeme-li chtít změnit nějakou jeho část, musíme změnu provést všude
- Použijeme funkci ✓

# Co je to funkce?

- Obecně se jedná o část programu, kterou je možné opakovaně “vyvolat” v různých místech programu.
-

# Struktura funkce

```
navratovy_typ jmeno(parametry) {  
    <implementace>  
}
```

- U funkce je třeba specifikovat:
  - návratový datový typ,
  - jméno (identifikátor),
  - parametry,
  - tělo (implementace).
- Funkce uvádíme mimo tělo funkce `main`.

# Funkce bez návratové hodnoty a parametrů

- Nejjednodušší typ funkce.
- Klíčové slovo `void`, prázdné závorky `()`

```
void greet() {  
    printf("Hello World!");  
}
```

- Samotná definice funkce nic nedělá  $\Rightarrow$  je třeba ji tzv. **zavolat**.

# Volání funkce

- Je třeba specifikovat, kde v programu se má daná funkce provést.

```
void greet() {  
    printf("Hello World!");  
}
```

```
int main() {  
  
    greet();    // Prints out "Hello World!"  
  
    return 0;  
}
```

# Otázky?

