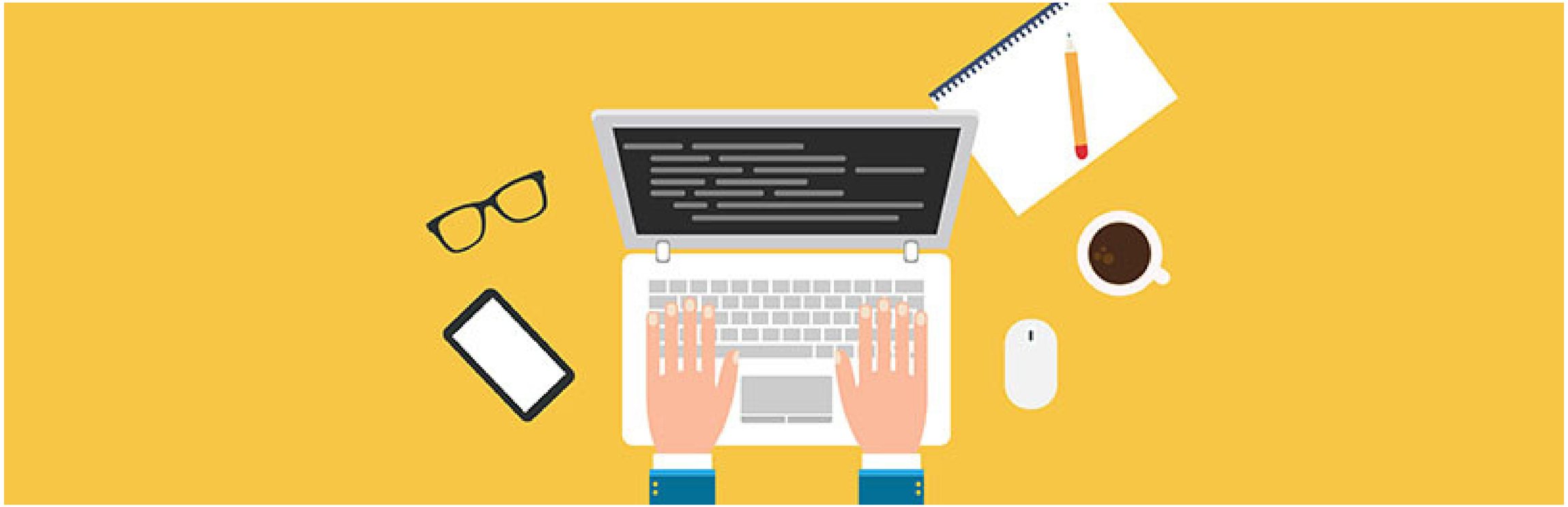

KURS PROGRAMIRANJA 2023

UVOD U PROGRAMIRANJE - MEDRESA CAZIN - 2023

FUNKCIJE

Funkcija je blok koda koji se pokreće samo onda kad je pozvan.

Funkciji se mogu proslijeđivati parametri.

Funkcije se koriste za izvođenje nekih akcija i važne su za ponovno korištenje radnji koje se ponavljaju (smanjuju kod).

KREIRANJE FUNKCIJE

C++ obezbjeđuje neke predefinirane funkcije kao što je main(). Ali i mi sami možemo praviti funkcije za određene akcije.

Da napravmo(deklariramo) funkciju, napišemo ime popraćeno sa zagradama ().

```
void myFunction() {
  // kod
}
```

POZIVANJE FUNKCIJE

```
// Kreiranje funkcije
void myFunction() {
  cout << "Hello";
}

int main() {
  myFunction(); // poziv funkcije
  return 0;
}

// "Hello"
```

DEKLARACIJA I DEFINICIJA

C++ funkcija se sastoji od dva dijela:

- DEKLARACIJA: tip(return tip), ime funkcije i parametri(ako ih ima)
- Definiocija: tijelo funkcije (kod koji se treba izvrsiti)

PARAMETRI I ARGUMENTI

Informacije se funkciji mogu proslijediti kao parametri. Parametri djeluju kao varijable unutar funkcije. Parametri se preciziraju poslije imena funkcije unutar zagrada.

```
void functionName(parameter1, parameter2, parameter3) {
    // kod
}
```

DEFAULTNI PARAMETAR

Moguće je koristiti i defaultnu verziju parametra, koristeći znak **=.**
To radimo ako pozivamo funkciju bez argumenta i u tom slučaju ona uzima defaultnu vrijednost.

```
void myFunction(string country = "Norway") {
 cout << country << "\n";
}
```

VIŠESTRUKI PARAMETRI

```
void myFunction(string fname, int age) {
 cout << fname << " Refsnes. " << age << " years old. \n";
}

int main() {
  myFunction("Liam", 3);
  myFunction("Jenny", 14);
  myFunction("Anja", 30);
  return 0;
}
```

VRAĆANJE VRIJEDNOSTI

Kad koristimo void ključnu riječ, kažemo da funkcije ne treba imati return vrijednost. Ako želimo da ima return vrijednost, koristimo neki tip podataka (int, string i sl.) umjesto void i koristimo ključnu riječ return unutar funkcije.

```
int myFunction(int x) {
  return 5 + x;
}
int main() {
 cout << myFunction(3);
  return 0;
}
```

VRAĆANJE VRIJEDNOSTI

```
int myFunction(int x, int y) {
  return x + y;
}

int main() {
  int z = myFunction(5, 3);
 cout << z;
  return 0;
}
```

PROSLIJEĐIVANJE PO REFERENCI

U prethodnim slučajevima koristili smo normalne varijable. Možemo proslijeđivati i reference.

Vidjeti primjer. numberswapwithreferences.cpp

PROSLIJEĐIVANJE NIZOVA

Moguće je proslijeđivati i nizove.

Vidjeti primjer:

arrayfunction.cpp

OVERLOADING FUNKCIJE

int myFunction(int x)
float myFunction(float x)
double myFunction(double x, double y)

REKURZIJA

Rekurzija je tehnika u kojoj funkcija zove sama sebe. Ova tehnika pomaže rješavanju složenih problema i smanjivanju istih na jednostavnije.

```
int sum(int k) {
  if (k > 0) {
    return k + sum(k - 1);
  } else {
    return 0;
  }
}

int main() {
  int result = sum(10);
  cout << result;
  return 0;
}
```

HVALA NA PAŽNJI

Next: Datoteke