

3.Ders : C Fonksiyon Türleri

C Programlamada Kullanıcı Tanımlı Fonksiyon Türleri

Örnek: Bağımsız değişken değeri *olan* ve dönüş değeri *olmayan* Fonksiyon

Örnek: Bağımsız değişken değeri ve dönüş değeri *olmayan* Fonksiyon

Örnek: Bağımsız değişkenli ve dönüş değeri *olmayan* Fonksiyon

Örnek: Değişkenler ve bir dönüş değeri olan Fonksiyon

Hangi yaklaşım daha iyidir?

Aşağıdaki bu 4 program, kullanıcı tarafından girilen tam sayının asal sayı olup olmadığını kontrol eder. Aşağıdaki tüm bu programların çıktısı aynıdır ve her örnekte kullanıcı tanımlı bir fonksiyon yarattık. Ancak, her örnekte yaklaşımımız farklı.

Örnek 1: Hiçbir argümanın iletilmediği ve geri dönüş değerinin olmaması durumu

```
#include <stdio.h>

void checkPrimeNumber();

int main()
{
    checkPrimeNumber();    // argument is not passed
    return 0;
}

void checkPrimeNumber()
{
```

```

int n, i, flag=0;

printf("pozitif bir tamsayı girin: ");
scanf("%d",&n);

for(i=2; i <= n/2; ++i)
{
    if(n%i == 0)
    {
        flag = 1;
    }
}
if (flag == 1)
    printf("%d bir asal sayı değildir.", n);
else
    printf("%d bir asal sayıdır.", n);
}

```

CheckPrimeNumber () fonksiyonu kullanıcıdan girdi alır, asal sayı olup olmadığını kontrol eder ve ekranda görüntüler.

Fonksiyonun dönüş tipi geçersiz. Bu nedenle, işlevden hiçbir değer döndürülmez.

Örnek 2: Hiçbir argümanın iletilmediği ancak geri dönüş değerinin olması durumu

```

#include <stdio.h>
int getInteger();

int main()
{
    int n, i, flag = 0;

    n = getInteger();    // no argument is passed

```

```

for(i=2; i<=n/2; ++i)
{
    if(n%i==0){
        flag = 1;
        break;
    }
}

if (flag == 1)
    printf("%d bir asal sayı değildir.", n);
else
    printf("%d bir asal sayıdır.", n);

return 0;
}

int getInteger()    // kullanıcı tarafından girilen tam sayıyı
döndürür
{
    int n;

    printf("pozitif bir tamsayı girin: ");
    scanf("%d",&n);

    return n;
}

```

N = getInteger () deyimi, işleve argüman iletilmediğini belirtir. Ve, fonksiyondan döndürülen değer n'ye atanır.

Burada, **getInteger ()** işlevi kullanıcıdan girdi alır ve onu döndürür. Bir sayının asal olup olmadığını kontrol etmek için gerekli kod main () işlevinin içindedir.

Örnek 3: Argümanların iletildiği ancak geri dönüş değerinin olmaması durumu

```

#include <stdio.h>
void checkPrimeAndDisplay(int n);

int main()
{
    int n;

    printf("pozitif bir tamsayı girin: ");
    scanf("%d",&n);

    // n is passed to the function
    checkPrimeAndDisplay(n);

    return 0;
}

// void indicates that no value is returned from the function
void checkPrimeAndDisplay(int n)
{
    int i, flag = 0;

    for(i=2; i <= n/2; ++i)
    {
        if(n%i == 0){
            flag = 1;
            break;
        }
    }
    if(flag == 1)
        printf("%d bir asal sayı değildir.",n);
    else
        printf("%d bir asal sayıdır.", n);
}

```

Kullanıcı tarafından girilen tamsayı değeri checkPrimeAndDisplay () işlevine geçirilir.

Burada, checkPrimeAndDisplay () işlevi, iletilen argümanın asal bir sayı olup olmadığını kontrol eder ve uygun mesajı

görüntüler.

Örnek 4: Argümanların iletildiği ve geri dönüş değerinin olması durumu

```
#include <stdio.h>
int checkPrimeNumber(int n);

int main()
{
    int n, flag;

    printf("pozitif bir tamsayı girin: ");
    scanf("%d",&n);

variable
    flag = checkPrimeNumber(n);

    if(flag == 1)
        printf("%d bir asal sayı değildir.",n);
    else
        printf("%d bir asal sayıdır.",n);

    return 0;
}

int checkPrimeNumber(int n)
{
    int i;

    for(i=2; i <= n/2; ++i)
    {
        if(n%i == 0)
            return 1;
    }

    return 0;
}
```

Kullanıcının giriři **checkPrimeNumber ()** fonksiyonuna iletilir.

CheckPrimeNumber () fonksiyonu, iletilen argümanın asal olup olmadığını kontrol eder. İletilen argüman bir asal sayıysa, fonksiyon 0 döndürür. İletilen argüman asal olmayan bir sayıysa, fonksiyon 1 döndürür. Dönüş deęeri bayrak deęişkenine atanır.

Bayrağın 0 mı yoksa 1 mi olduğuna baęlı olarak, main () fonksiyonunda uygun bir mesaj yazdırılır.

Hangi yaklaşım daha iyidir?

Çözmeye çalıştığınız soruna baęlı. Çoęu durumda, argüman iletme ve işlevden bir deęer döndürme (**örnek 4**) daha iyidir.

Bir fonksiyon belirli bir görevi yerine getirmelidir. CheckPrimeNumber () işlevi kullanıcıdan girdi almaz, sadece bir sayının asal olup olmadığını kontrol eder, bu kodunuzun anlaşılmasını ve hata ayıklamasını kolaylaştırır.