

## Naming Conventions (İsimlendirme kuralları)

- Harf veya alt tire ( \_ ) (underscore) ile başlar
- Harf, rakam veya alt tire ( \_ ) kombinasyonlarını içerebilir
- İsmi karakter sayısı 32'den az olmalıdır. (bazı derleyiciler fazlasını destekleyebilir)
- C harfi **keyword** değildir.

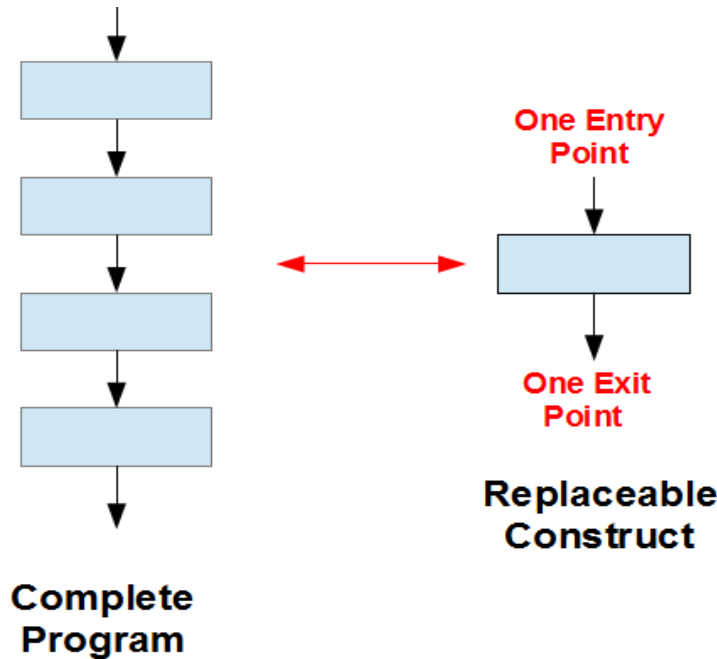
## Değişkenleri İyi İsimlendirme Teknikleri

### Değişken isimleri (**identifiers**)

- Kendi kendini açıklayabilir olmalı. Ne amaçla kullanıldıklarını açıklamak için yorum gerektirmemeli (Bu hiç yorum yapılmayacağı anlamına gelmez)
- Özlü olmalı ama şifreli olacak kadar kısa değil
- Depolanan veriyi doğru bir şekilde tanımlamalı
- Kodun okunmasını kolaylaştırmaya yardımcı olmalı
- Biz **alt\_tire** ( \_ ) notasyonunu kullanacağız. "test\_value"
- İstenirse **macar** notasyonu veya **camelCase** notasyonu da kullanılabilir. "intTestValue" veya "testValue"

## Structured Programming (Yapısal Programlama)

Yapısal programlama, her birinin bir giriş noktası ve bir çıkış noktası olan yapı kümelerinden oluşur. Herhangi bir programcı, programdaki diğer yapıları etkilemeden veya hatalara (**bugs**) yol açmadan bir yapıyı değiştirebilir, geliştirebilir, başka geliştirilmiş bir yapıyla değiştirebilir.



# Workshop #3

## Ödevi Zamanında Verememe/Eksik Ödev Teslimi

- Bir Paket 10'lu kahve getirilecek.
- Kullanıcıdan aldığı sayıya kadar olan fibonacci sayı dizisini ekrana basacak ve bütün satırları yorumlayacak.

## Giriş

Bu çalışmada verilerinizi uygun tipteki değişkenlerde saklayacak, bu değişkenler üzerinde hesaplamalar yapacak, ve çeşitli test senaryolarının **true** veya **false** sonucunu belirlemek için ilişkisel (**reletional**) ve mantıksal ifadeleri (**logical expression**) kullanacaksınız.

## Konular

- Veri tipleri
- Sabitler ve değişmezler.
- Mantıksal ve karşılaştırma operatörleri

## Beklenti

- Program değişkenlerini ve sabitelerini depolamak için uygun türleri seçme
- Okunması ve bakımı kolay, koda katkıda bulunacak anlamlı değişken isimleri oluşturabilme
- Temel bir programlama görevini çözmek için basit hesaplamaları kodlayabilme
- Çeşitli test ifadelerinin doğru veya yanlış sonuçlarını değerlendirmek için ilişkisel ve mantıksal ifadeleri (**reletional and logical expression**) kodlayabilme

---

## Part-1 (%10)

Zip dosyanızı bilgisayarınızda Cdersleri\Workshops\ klasörünüze çıkarın. WS03 klasörünü VSC editörünüz ile proje olarak açın.

1. "Part-1 Output Example" kısmında programın çıktılarını, programın nasıl çalışacağını anlamak için dikkatlice inceleyin.
2. Kodlarınızı proje klasörünüzdeki p1 klasöründe bulunan "**w3p1.c**" dosyasına verilenleri değiştirmeden yazın.
3. Bu program birkaç ana mantıksal kod bölümüne ayrılır:
  - a. **Variable declaration:**  
Bütün değişkenlerin ana fonksiyonun (**main** function) içinde en üstde ve anlamlı gruplar halinde bildirildiği bölüm
  - b. **Display product data:**  
Verilerin okunmasını kolaylaştırmaya yardımcı olmak için ürün verilerinin tablo biçiminde özetlendiği bölüm
  - c. **Display summary of test scenarios:**  
Her test senaryosunda sonuçların **true** (non-zero) ve **false** (zero) temsilini üretmek için ilişkisel ve mantıksal ifadelerin kullanıldığı bölüm.

4. Verilen “test\_value” değişkenini silmeyin veya değiştirmeyin.
5. Aşağıdaki verileri saklamak için 9 adet değişkene ihtiyacınız olacak. Üç ürünün her biri için üçer tane değişken tanımlamanız gerekecek.

	Product-1	Product-2	Product-3
Product ID	111	222	111
Price	111.49	222.99	334.49
Taxed	Y	N	N

#### Not

- Tüm değişkenler değiştirilemez(**unmodifiable** or **readonly**) olduğundan değişken bildirimlerinde programın akışı boyunca bu değişkenlerin değiştirilemeyeceğini garanti altına alacaksınız.
  - Saklanması gereken veri tipine göre her değişken için uygun veri tipini seçmelisiniz.
  - Kodun okunabilirliğini ve sürdürülebilirliğini en üst düzeye çıkarmak için kendi kendini açıklayan değişken isimleri kullanmalısınız. ( kodu okuyan kişi bu değişkeni ne amaçla kullandın diye sormamalı)
6. Üç ürünün ortalama fiyatını saklayacağınız bir değişkene daha ihtiyacınız olacak. Üç ürünün ortalamasını hesaplayıp bu değişkeni öyle başlatacaksınız. (**declaration** değil **initialization** yapacaksınız.)
  7. Bildirilmiş (**declared**) ve başlatılmış (**initialized**) toplam 11 değişkenin olması gerekiyor. Daha fazla değişken kabul edilmeyecek.
  8. Her ürün bilgisini “Part-1 Output Example” kısmındaki örnek çıktıdaki gibi görüntüleyin.
    - **SARI** olarak vurgulanan kısımlarda sizin tanımladığınız sabitlerin (**const** variables) değerleri printf fonksiyonuna formatlı veri olarak verilecek. (tablodaki değerler printf içine sabit olarak elle yazılmayacak)
    - Fiyatların precision değerlerine dikkat edin. Noktadan sonra 4 basamak görüntülenecek.
  9. Üç ürünün hesaplanan ortalama değerini örnek çıktıdaki gibi ekrana basın. Yine precision değeri 4 olacak ve bu fiyatı ekrana basmak için initialize ettiğiniz değişkeni kullanacaksınız.
  10. İlişkisel ve mantıksal ifadeler ile ilgili 5 satırlık bilgiyi örnek çıktıdaki gibi ekrana basın. Biçimlendirmeye dikkat edin
  11. Örnek çıktıdaki gibi 9 veri analizi bölümünü görüntüleyin.
    - Bu bölümdeki bütün fiyatların precision değeri noktadan sonra iki basamaktır.
    - **SARI** vurgulanan kısımlar printf içinde değişkenleri nerede kullanmanız gerektiğini gösterir.
    - **YEŞİL** vurgulanan kısımlar ilişkisel ve/veya mantıksal bir ifadenin sonucunu temsil eder. Gerekli sonucu elde etmek için uygun değişkenleri uygun ilişkisel ve/veya mantıksal operatörler ile birlikte kullanmalısınız. (printf’de ekrana basılan açıklamaya dikkat edin)
    - **KIRMIZI** vurgulanan kısım bir satır içi hesaplamanın sonucunun nereye yazılacağını belirtir. Bu hesaplama bir değişkene kaydedilemez.

#### Not

- Derlenen dosyaya Windows sistemlerde “prg\_w3p1.exe”, Linux sistemlerde “prg\_w3p1” adı verilecek.
- Derleyici parametrelerinden **-Wall** ve **-Werror** kullanılacak.

## Part-1 Oputput Example

### Product Information

=====

Product-1 (ID:111)

Taxed: Y

Price: \$111.4900

Product-2 (ID:222)

Taxed: N

Price: \$222.9900

Product-3 (ID:111)

Taxed: N

Price: \$334.4900

The average of all prices is: \$222.9900

### About Relational and Logical Expressions!

=====

1. These expressions evaluate to TRUE or FALSE
2. FALSE: is always represented by integer value 0
3. TRUE : is represented by any integer value other than 0

### Some Data Analysis...

=====

1. Is product 1 taxable? -> 1
2. Are products 2 and 3 both NOT taxable (N)? -> 1
3. Is product 3 less than test\_value (\$330.99)? -> 0
4. Is the price of product 3 more than both product 1 and 2 combined? -> 1
5. Is the price of product 1 equal to or more than the price difference of product 3 LESS product 2? -> 0 (price difference: \$111.50)
6. Is the price of product 2 equal to or more than the average price? -> 1
7. Based on product ID, product 1 is unique -> 0
8. Based on product ID, product 2 is unique -> 1
9. Based on product ID, product 3 is unique -> 0

## **Part-2 (%40)**

1. Programın nasıl çalıştığına dair fikir edinmek için “**Part-2 Output Example**” kısmını dikkatlice inceleyin.
2. Çözümünüzü p2 klasöründe yer alan **w3p2.c** dosyasına kodlayın.
3. Bu program birkaç ana mantıksal kod bölümüne ayrılabilir:
  - a. **Variable declaration:** Bütün değişkenlerin main fonksiyonunun en üstünde ve anlamlı gruplar halinde bildirildiği bölüm
  - b. **Product data input:** Üç ürünle ilgili bilgileri kullanıcıdan alıp, uygun değişkenlere yerleştirdiğiniz bölüm
  - c. **Display product data:** Verilerin okunmasını kolaylaştırmak için ürün verilerini tablo şeklinde özetlediğiniz bölüm
  - d. **User preference input:** Kullanıcının kahve tercihlerini sorup, kullanıcı tercihlerini uygun değişkenlerde sakladığınız bölüm
  - e. **Display summary of results:** Ürün bilgilerinin kullanıcı tercihleriyle nasıl eşleştiğinin belirlendiği bölüm
  - f. **Repeat:** başka bir tercih senaryosu için “d” ve “e” bölümlerinin tekrarlandığı bölüm
4. Size verilen “GRAMS\_IN\_LBS” değişkeninin bildirimini silmeyin veya değiştirmeyin. Gram ağırlık türünden **pounds** ağırlık türüne dönüşüm sırasında buna ihtiyacınız olacak.
5. Örnek çıktıyı klavuz olarak kullanarak üç ürün verisini kaydetmek için kullanıcıdan alıp kaydedeceğiniz onbeş (15) değişkenin bildirimini yapın.

### **Not**

- Tanımlayacağınız bu değişkenlerin veri tiplerini saklayacağınız verilerin tipine göre seçmek sizin sorumluluğunuzda
  - Kodun okunabilirliğini ve sürdürülebilirliğini en üst düzeye çıkarmak için **self-describing** (kendi kendini açıklayan) isimlendirme tekniğini kullanmalısınız.
6. Kullanıcıdan tek karakterlik veri alırken programımız anlam veremediğimiz bir davranış sergileyebilir. Bundan kaçınmak için şimdilik derste öğrendiğimiz aşağıdaki scanf formatını kullanabilirsiniz.

```
scanf(" %c", ...
```

7. Ürün verilerinin tablo şeklinde görüntülenmesi, biraz daha gelişmiş bazı biçimlendirme özelliklerinin uygulanmasını gerektirir. Bununla ilgili araştırma yapmanız gerekebilir. Şimdilik başlamak için aşağıdaki ilk ürün verisi satırını kullanabilirsiniz.

```
printf(" 1 |   %d   |   %d   |   %d   |   %d   | %4d | %6.3lf |   %d   | %5.1lf | %5.1lf\n", ...
```

8. Aşağıdaki tablo, her bir ürünün kullanıcı giriş tercihlerini eşleştirmek için uygulamanız gereken eşleme kurallarını gösterir. (Örneğin kullanıcı “Mild” kahve tercih ederse ve ürün tipi de “Light” ise bu durum tabloda ilgili kısımda **true** (non-zero) olarak işaretlenecek. Elbette bu durumu karşılaştırma operatörleriniz belirleyecek):

Preference	Product
Coffee Strength	Coffee Type
Mild	Light
Rich	Blend
Coffee Making Equipment	Coffee Grind Size
Residential	Course
Commercial	Fine
Daily Servings <i>(inclusive range)</i>	Coffee Package Weight <i>(inclusive range)</i>
1 to 4	0 to 250 g
5 to 9	500 g
10 or more	1000 g
Like Cream with Coffee	Suggest Serving with Cream
Yes	Yes
No	No
Coffee Making Equipment	Serving Temperature <i>(inclusive range)</i>
Residential	60.0 to 69.9 (°C)
Commercial	70.0 or more (°C)

9. Kullanıcı tercihlerini ve kahve yapımında kullanılan ekipmanın bir senaryosunu açıklayan kullanıcı girdisi değerlerini saklamanız için gereken değişken sayısı 4 (dört)’den fazla olmayacaktır. Bu değişkenler girdi senaryosuna göre her bir ürünün ihtiyaçları nasıl karşıladığını belirlemede kullanılacaktır (yukarıdaki tabloya göre) .
10. Tercih edilen ürün özeti sonuçlarının tablo biçiminde görüntülenmesi için kullanmanız gereken printf formatı aşağıda verilmiştir:

```
printf(" 1|      %d      |      %d      |      %d      |      %d      |      %d\n", ...
```

#### Not

Bu formata yerleştirmek için kullanacağınız değişkenleri uygun ilişkisel ve mantıksal ifadelerle (**relational and/or logical expression(s)**) birlikte kullanmanız gerekecektir.

#### Not

- Derlenen dosyaya Windows sistemlerde “**prg\_w3p2.exe**”, Linux sistemlerde “**prg\_w3p2**” adı verilecek.
- Derleyici parametrelerinden **-Wall** ve **-Werror** kullanılacak.

## Part-2 Output Example (Not: SARI alanlar kullanıcı veri girişini temsil eder)

Take a Break - Coffee Shop

=====

Enter the coffee product information being sold today...

COFFEE-1...

Type ([L]ight,[B]lend): **l**

Grind size ([C]ourse,[F]ine): **c**

Bag weight (g): **250**

Best served with cream ([Y]es,[N]o): **y**

Ideal serving temperature (Celsius): **65.7**

COFFEE-2...

Type ([L]ight,[B]lend): **B**

Grind size ([C]ourse,[F]ine): **F**

Bag weight (g): **500**

Best served with cream ([Y]es,[N]o): **N**

Ideal serving temperature (Celsius): **70.0**

COFFEE-3...

Type ([L]ight,[B]lend): **L**

Grind size ([C]ourse,[F]ine): **f**

Bag weight (g): **1000**

Best served with cream ([Y]es,[N]o): **n**

Ideal serving temperature (Celsius): **80.5**

-----+-----+-----+-----+-----+-----										
	Coffee		Coffee		Packaged		Best		Serving	
	Type		Grind Size		Bag Weight		Served		Temperature	
	+-----+-----		+-----+-----		+-----+-----		+ With		+-----+-----	
ID	Light	Blend	Course	Fine	(G)	Lbs		Cream	(C)	(F)
+-----+-----+-----+-----+-----+-----										
1	1	0	1	0	250	0.551		1	65.7	150.3
2	0	1	0	1	500	1.102		0	70.0	158.0
3	1	0	0	1	1000	2.205		0	80.5	176.9

Enter how you like your coffee and the coffee maker equipment you use...

Coffee strength ([M]ild,[R]ich): **m**

Do you like your coffee with cream ([Y]es,[N]o): **y**

Typical number of daily servings: **6**

Maker ([R]esidential,[C]ommerical): **c**

The below table shows how your preferences align to the available products:

ID	Coffee Type	Coffee Grind Size	Packaged Bag Weight	With Cream	Serving Temperature
1	1	0	0	1	0
2	0	1	1	0	1
3	1	1	0	0	1

Enter how you like your coffee and the coffee maker equipment you use...

Coffee strength ([M]ild,[R]ich): R

Do you like your coffee with cream ([Y]es,[N]o): N

Typical number of daily servings: 10

Maker ([R]esidential,[C]ommercial): R

The below table shows how your preferences align to the available products:

ID	Coffee Type	Coffee Grind Size	Packaged Bag Weight	With Cream	Serving Temperature
1	0	1	0	0	1
2	1	0	0	1	0
3	0	0	1	1	0

Hope you found a product that suits your likes!