

Algoritma ve Programlamaya Giriş

Giriş ve Temel Kavramlar

1. Yarıyıl

Hafta-01

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse giriş niteliğinde olan konulara değinilecek ve Algoritma sözcüğü ve kavramları hakkında temel bilgiler verilecektir.

Haydi başlıyoruz...

****Musa El Harizminin Resmi konacak****

Algoritma ve Programlamaya Giriş

Sizin de bildiğiniz gibi bir bilgisayar sistemi donanım ve yazılım olmak üzere iki temel bileşenden oluşmaktadır. Donanım, bilgisayarın gözle görülebilen fiziksel parçalarının tamamıdır. Yazılım ise, bilgisayarın donanımını çalıştıran, soyut bileşenlerdir.

Bir bilgisayarın yazılım sistemi ise, sistem yazılımları, geliştirme yazılımları ve uygulama yazılımları olmak üzere üç farklı bileşenden oluşmaktadır.

Buraya bir bilgisayar resmi eklememiz gerekiyor.

**Bilgisayar donanım ve yazılım parçalarının resimleri konacak **

Algoritma ve Programlamaya Giriş

Sistem yazılımları, bilgisayarın donanımına en yakın olarak çalışan ve donanım işlevlerinin doğru ve yüksek başarımlı olarak çalışmasını sağlayan yazılım bileşenidir. Bunlara, en iyi örnek işletim sistemi (operating system) yazılımlarıdır. En iyi bilinen işletim sistemleri ise, Microsoft firmasının Windows'u, IBM firmasının AS/400'ü, Sun firmasının Solaris'i ve 'ortak akılla' tamamen açık kaynak kod olarak gerçekleştirilen Linux olarak verilebilir.

Buraya bir bilgisayar resmi eklememiz gerekiyor.

**Bilgisayar donanım ve yazılım parçalarının resimleri konacak **

Algoritma ve Programlamaya Giriş

Algoritma, herhangi bir sorunun çözümü için izlenecek yol anlamına gelmektedir. Komutları programlama dillerinin komut yapılarına benzetilerek kurgulandığından, algoritma kavramlarının öğrenilmesi, bilgisayar programlarının çalışma mantığının da daha rahat kavranılmasını sağlar. Herhangi bir programlama dili eğitimi almadan önce algoritmik bir çalışma sürecinden mutlaka geçilmesi gerekmektedir.

Düşünen ve/veya 0 ve 1 lerle çalışan bir makine/insan resimleri konabilir.

Algoritma ve Programlamaya Giriş

Bir anlamda, çözücülük olarak da, adlandırabileceğim algoritma çalışmalarının , bunu destekleyecek her türlü bilgiye de gereksinimi olduğunu söyleyebilirim. Bu konuda hedefleri olan sizlerin,yazılım sektöründe hakettiğiniz bir yerde olabilmeniz için , mutlaka matematik, felsefe, genel kültür vb. gibi konularla ilgili bilgilerinizi de zenginleştirmeniz gerekir. Algoritmik yaklaşım, yani çözücü yaklaşımınızı güçlendirmeniz, sizi programlama tekniklerine de yaklaştırmış olacaktır.

Algoritma kutularından serpiştirelim.

Algoritma ve Programlamaya Giriş

Bu derse ilişkin notlarımı, programlama dili eğitimi öncesi problem çözme sistematığını ve programlamanın temel mantığını kavrama amacına yönelik hazırladım.

Çözüm için yapılması gereken işlemler, hiçbir alternatif yoruma izin vermeksizin sözel olarak ifade edilir. Diğer bir deyişle algoritma verilerin bilgisayara hangi çevre biriminden, girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağıının, sonucun nasıl ve nereye yazılacağıının sözel olarak ifade edilmesi biçiminde tanımlanabilir

Harizmi nin resmini tekrar koyabiliriz.

Algoritma ve Programlamaya Giriş

Tüm ürünlerimizin Bilgisayar Teknolojileri içerisinde hayata geçirileceği gerçeğinden hareketle,Algoritmik çalışmalar (çözömlmeler), sonuçlandıktan sonra doğal olarak , donanım üzerinde çalışan bir YAZILIM ortaya çıkacaktır. Tanımlama olarak YAZILIM;

Değişik ve çeşitli görevler yapma amaçlı tasarlanmış elektronik araçların, birbirleri ile haberleşebilmesini ve uyumunu sağlayarak, görevlerini ya da kullanılabilirliklerini geliştirmeye yarayan makine komutlarıdır. Yazılım, elektronik cihazların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir.

0 ve 1 lerden oluşan ilginç bir görsellik kullanılabilir.

Algoritma ve Programlamaya Giriş

YAZILIM için, bilgisayarın RUH udur dersek ilginç ama doğru bir benzetme yapmış oluruz. Bu durumda bilgisayarın donanım kısmına da , BEDEN benzetmesi yapabiliriz.

Yani bilgisayarın donanımı üzerinde planlanan işlerin gerçekleştirilmesini YAZILIM larımız yaparlar.

Örneğin, şirketimizin çalışanlarının kayıtlarını tutmak istiyorsak bir programa ihtiyaç var demektir. Bu isteğimizi farklı şekillerde, farklı yazılımlar aracılığı ile gerçekleştirebiliriz.

Ancak , özel çözümler bulmak için program yazmak ya da uygulama geliştirmek söz konusu olacaktır.

Bir programcı resmi olabilir.

Algoritma ve Programlamaya Giriş

Algoritmalar programlama dillerinden bağımsız , çözücü çalışmalardır. Ancak yine de her yazılımcının ustalaştığı bir programlama dili vardır ve algoritmik çalışmalarını doğal olarak usta olduğu programlama dili üzerinde gerçekleştireceğini düşünerek yürütür.

Programlar, öncelikle bir işletim sistemi için yazılır. Windows XP için yazılan bir program Linux işletim sistemi için çalışmaz.

Bir programcı resmi olabilir.

Algoritma ve Programlamaya Giriş

Programlama Dili Kodlarına İlişkin Örnekler

C Programlama Dili

```
alis_fiyati = 10
```

```
satis_fiyati = 20
```

```
kar = satis_fiyati – alis_fiyati
```

```
if (kar > 0) ...
```

Assembly Dili

```
MOV byte ptr alis_fiyati, 10
```

```
MOV byte ptr satis_fiyati, 20
```

```
SUB satis_fiyati, alis_fiyati
```

```
MOV byte ptr kar, byte ptr satis_fiyati
```

```
CMP kar, 0
```

```
JA kar_var
```

İkisi de aynı amacı gerçekleştiren kodların birbirlerinden ne kadar farklı olduğunu görüyoruz.

Algoritma ve Programlamaya Giriş

İşletim Sistemi tanımı olarak şunları yazabiliriz.

“İşletim Sistemi, bilgisayar donanımının doğrudan denetimi ve yönetiminden, temel sistem işlemlerinden ve uygulama programlarını çalıştırmaktan sorumlu olan sistem yazılımıdır.”

Unix, akademik çevrelerde ve sunucu olarak kullanılmakta iken, Windows ise evde ve ofislerde masaüstünde tercih edilmektedir.

Anabilgisayarlar ve gömülü sistemler ise çoğu Windows ve Unix ile doğrudan bağlantısı olmayan pek çok değişik işletim sistemi kullanılmaktadır.

(Vikipedi, Özgür Ansiklopedi)

Farklı bilgisayar uygulamalarına ilişkin resimler olabilir

Algoritma ve Programlamaya Giriş

Programlama dilleri kendi aralarında sınıflara ayrılmışlardır. Yüksek ya da orta düzeyli programlama dilleriyle, düşük düzeyli programlama dilleri bazı yönlerden karşılaştırılabilir.

Hız: yüksek ya da orta düzeyli programlama dillerinde yazılan bir deyim, çalıştırılabilmek için birden fazla makina dili deyimine çevrileceğinden, çalışma hızı yavaştır. Birleştirici dilinde yazılmış deyimlerin herbiri bir tek makina dili deyimine karşılık geldiğinden, birleştirici dili ile yazılmış programların çalışma hızları daha yüksektir.

Farklı bilgisayar uygulamalarına ilişkin resimler olabilir

Algoritma ve Programlamaya Giriş

Program Hacmi: yüksek ve orta düzeyli programlama dilleri ile yazılan programların hacimleri, düşük düzeyli programlama dili ile yazılan programlardan daha küçüktür (ben burada programın hacmini, program içerisinde verilen satır sayısı olarak belirtiyorum, buna sizinde bildiğiniz gibi kaynak kod –source code- adı da verilir).

Donanım Bağımlılığı: düşük düzeyli programlama dilleri ile geliştirilen programlar, sistem donanımı ile doğrudan ilişki içerisinde buldukları için diğer bir donanıma taşınmaları mümkün olmamaktadır. Burada, farklı donanımları, örneğin, Intel Pentium işlemcisi tabanlı bir sistem ile, Motorola tabanlı bir donanım olarak düşünebilirsiniz.

Farklı bilgisayar uygulamalarına ilişkin resimler olabilir

Algoritma ve Programlamaya Giriş

Ayrıca, gerçekleştirilecek işlevlerin sadece donanımla ilintili olmadığını, donanım, işletim sistemi (işletim sisteminin sürümü) gibi bileşenlerin tamamının etkin olduğunu düşünmek gerekmektedir. Donanım ile işletim sisteminin tamamına platform adı verilir. Yüksek düzeyli programlama dillerine bakıldığında, bunların birçoğunun bazı kısıtlamalar altında platform bağımsız sonuçlar verebileceğini düşünmek gerekmektedir. Burada, platform bağımsızlıktan kasıt, yazılan bir programın, sistemin donanımı ve işletim sistemi değiştirildiğinde bile çalışıp, aynı sonuçları vermesidir.

Farklı bilgisayar uygulamalarına ilişkin resimler olabilir

Algoritma ve Programlamaya Giriş

Tabii ki, burada, düşük düzeyli programlama dilleri de çok önemli bir yer tutmaktadır. Yüksek düzeyli programlama dillerinin, sadece yüksek düzeyli uygulamalar geliştirmek için kullanıldığını unutmayın. Bu uygulamalar, genellikle çok yüksek donanım kaynağı isteyen (hızlı işlemciler, megabyte'larca ana bellek, güçlü bir giriş çıkış altyapısı gibi) bir yapı gösterirler. Kullandığınız işletim sistemlerinin ve uygulama yazılımlarının istedikleri kaynakları tararsanız bunları çok daha iyi anlayabilirsiniz. Düşük düzeyli programlama dillerine döndüğümüzde ise, bunların çoğunlukla, çok düşük işlemci hızları isteyen, az miktarda bellek üzerinde çalışan uygulamalar olduğunu görürüz. Bunlara genellikle gömülü (embedded) uygulamalar adı verilir.

Farklı bilgisayar uygulamalarına ilişkin resimler olabilir

Algoritma ve Programlamaya Giriş

Gömülü uygulamalara en iyi örnekler, çamaşır makinalarımızı, televizyonlarımızı, VCD oynatıcılarımızı çalıştırdığımız uygulamalardır. Şimdi, bir düşünün. Az önce saydığımız kaç tane aygıtın içerisinde, bir Pentium işlemci ve 1 Gbyte bellek bulunur?

Farklı bilgisayar uygulamalarına ilişkin resimler olabilir

Algoritma ve Programlamaya Giriş

Programlama dilleri esas olarak kendi aralarında,

- a) I. Kuşak, b) II. Kuşak, c) III. Kuşak
- d) IV. Kuşak diye 4 ana gruba ayrılırlar.

I. Kuşak diller , makine dili komutlarından oluşan bir programlama idi. Bu “assembly” dilinin oluşmasına yol açmıştır.

Bu dili kullanmak gerçekten çok zordur, hata yapma riski yüksektir.

ATM Makinalarından bir resim olabilir.

Algoritma ve Programlamaya Giriş

II.Kuşak diller , alçak seviyeli diller olarak söylenebilir. Makine dillerinden sonra ciddi bir rahatlık getirmiş olsa da yine de kodlamanın zorluk derecesi yüksektir.

Orta seviyeli bir dilde 10-15 satırlık kod ile halledilen bir mantık çözümü, burada 80-90 satırlık bir kodlama ile çözülür.

Ancak, çalışma hızı yüksektir. Donanım ile ilgili kodlamalarda tercih edilmesi daha uygun olur.

Bir bilgisayarın anakart resmi olabilir.

Algoritma ve Programlamaya Giriş

III.Kuşak diller , Yüksek Seviyeli Dillerdir. Yapısal programlamanın öne çıktığı ve assembly dilinin zorluklarını aşmaya yönelik dillerdir.

Fortran, Cobol, Algol, Pascal, PLI, C, C++ dillerini bu kategoride sayabiliriz. Halen günümüzde C, C++ çok yaygın olarak kullanılmaktadır.

Bu dillere fonsiyonlar dili de denilebilir. İş yapan mantıkları fonsiyonlar bazında paketleme olanağı verirler. Fonsiyonların, daha sonra farklı çözümlmelerde de kullanılmaları çok önemli bir aşamadır.

Bir bilgisayarın anakart resmi olabilir.

Algoritma ve Programlamaya Giriş

IV.Kuşak diller , Çok Yüksek Seviyeli Dillerdir. Sadece ne istediğimizi bildirdiğimiz, nasıl yapacağımızı söylemediğimiz programlama dilleridir.

LISP, SNOBOL, FORTH, SQL bu dillere örnek olarak verilebilir.

Bunların arasında çok fazla kullanılan, ilişkisel veri tabanlarının sorgulama dili olan SQL ile, yapay zeka programlama dili olan LISP tir.

Bir bilgisayarın anakart resmi olabilir.

Algoritma ve Programlamaya Giriş

Uygulama Alanları açısından bakılacak olursa;

Pascal, C, C++, bilimsel kuruluşlarda ve mühendislik veya matematik hesapları için kullanılır. Fortan programlama dili halen bir çok üniversitenin mühendislik bölümlerinde ders olarak okutulmakta olsa da güncelliğini yitirmiştir.

Farklı animeler olabilir.

Algoritma ve Programlamaya Giriş

Uygulama Alanları açısından bakılacak olursa;

SQL (Structured Query Language)
Yapılandırılmış Sorgulama Dili , Veritabanı kullanımında yaygın olarak kullanılan bir dildir. Günümüzde bütünleşik bir çok veritabanı yönetim sistemi olsa da hepsinin temelinde SQL bulunmaktadır.

Farklı animeler olabilir.

Algoritma ve Programlamaya Giriş

Uygulama Alanları açısından bakılacak olursa;

C, C++, Java ve Makine Dilleri , İşletim Sistemlerinin ve Sistem programlarının yazılımında kullanılırlar.

Genel Amaçlı olarak, çeşitli uygulamalar geliştirmek için kullanılan dillere örnek olarak da, yine C,C++, Java, VB ve Pascal dillerini verebiliriz.

Farklı animeler olabilir.

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar bařlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, nceki slaytları tekrar inceleyiniz.

Cevapları bir sonraki hafta sonraki notlarımızın bařında bulabilirsiniz.

Çözümlü Sorular

....., bilgisayarın gözle görülebilen fiziksel parçalarının tamamıdır.

Noktalı yere hangi ifade gelecektir ?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Algoritmik çalışmalar (çözümlemeler), sonuçlandıktan sonra doğal olarak , donanım üzerinde çalışan bir ortaya çıkacaktır.

Noktalı yere hangi ifade gelecektir ?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

....., bilgisayarın donanımını çalıştıran, soyut bileşenlerdir.

Noktalı yere hangi ifade gelecektir ?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

....., elektronik cihazların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir.

Noktalı yere hangi ifade gelecektir ?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

....., programlama dillerinden bağımsız , çözücü çalışmalarıdır.

Noktalı yere hangi ifade gelecektir ?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programlama dilleri esas olarak kendi aralarında,
Hangi guruplara ayrılır?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Veritabanı kullanımında yaygın olarak kullanılan programlama dili hangisidir?

** Şıklar **

Yanıt gelecek derste verilecek.

Çözömlü Sorular

İşletim Sistemlerinin ve Sistem programlarının yazılımında hangi programlama dilleri kullanılırlar ?

** Şıklar **

Yanıt gelecek derste verilecek.

Ders Materyali

Bu dersle ilgili materyal çok miktarda ve ayrıntılı olmakla birlikte, başlangıç için ařağıdaki iki kitabı önerebilirim.

Herbert SCHILDT, C Complete Reference ALFA Yayınları

Ders Sonu

Bu dersle birlikte programcılar dünyasına ilk adınızı attınız.

Daha fazla bilgiyi gelecek derslerde bulabilirsiniz.

Hoşçakalın.

Başarılar.

Algoritma ve Programlamaya Giriş

Programlama Yapmak

1. Yarıyıl

Hafta-02

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse programlama yapma ile ilgili konulara değinilecek ve Algoritma sözcüğü ve kavramları hakkında temel bilgiler verilecektir.

Haydi başlıyoruz...

****Musa El Harizminin Resmi konacak****

Çözümlü Sorular

....., bilgisayarın gözle görülebilen fiziksel parçalarının tamamıdır.

Noktalı yere hangi ifade gelecektir ?

** Donanım **

Çözümlü Sorular

Algoritmik çalışmalar (çözümlemeler), sonuçlandıktan sonra doğal olarak , donanım üzerinde çalışan bir ortaya çıkacaktır.

Noktalı yere hangi ifade gelecektir ?

** Yazılım **

Çözömlü Sorular

....., bilgisayarın donanımını çalıştıran, soyut bileşenlerdir.

Noktalı yere hangi ifade gelecektir ?

** Yazılım **

Çözömlü Sorular

....., elektronik cihazların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir.

Noktalı yere hangi ifade gelecektir ?

** Yazılım **

Çözümlü Sorular

....., programlama dillerinden bağımsız , çözücü çalışmalarıdır.

Noktalı yere hangi ifade gelecektir ?

**** Algoritmalar ****

Çözümlü Sorular

Programlama dilleri esas olarak kendi aralarında,
Hangi guruplara ayrılır?

** I. , II. , III. , IV. Kuşak Programlama Dilleri **

Çözümlü Sorular

Veritabanı kullanımında yaygın olarak kullanılan programlama dili hangisidir?

**** SQL ****

Çözömlü Sorular

İşletim Sistemlerinin ve Sistem programlarının yazılımında hangi programlama dilleri kullanılırlar ?

** C, C++, Java ve Makine Dilleri **

Algoritma ve Programlamaya Giriş

Bilgisayar programı dediğimiz zaman öncelikle, bilgisayara ne yapması gerektiğini söyleyen makine komutlarından söz ediyoruz demektir. Önceki hafta verdiğimiz programlama dillerinin hemen hepsinin de , bilgisayarın anlayacağı komutlara dönüştürülmesi gerekmektedir.

Bu işleme Compile (derleme) veya Interprinting (yorumlama) denir.

Uygun bir resim konacak

Algoritma ve Programlamaya Giriş

Bilgisayar programı yazılımında makro düzeyde 4 aşama bulunur. Bunlar;

- A)Analiz Aşaması
- B)Algoritma Aşaması
- C)Programlama Dilinin Seçilmesi
- D)Kodlama Aşaması

Uygun bir resim konacak

Algoritma ve Programlamaya Giriş

A)Analiz Aşaması

Bilgisayar ortamında çözülecek olan sorunun, uygulama haline dönüştürülmesi için yapılan ve sorunun genel hatlarının ortaya çıkarıldığı aşamadır. Tasarımlamanın ve ilgili parametre değişkenlerinin belirlenmesi gerçekleşir. Tabii bunlar tamamen, uygulamayı kullanacak olan son kullanıcılar ile yapılacak ortak toplantılar ile belirlenecektir. Olabilecek sonuçlar ve/veya kısıtlar daha çok metin ifadelere dökülür.

Masa başında toplanan insanları yansıtan bir resim olabilir.

Algoritma ve Programlamaya Giriş

B) Algoritma Aşaması

Yapılan analiz çalışması sonucunda sorunu çözüme götürecek yol belirlendiğinde algoritma oluşmuş olur. Okula ilk defa gelecek bir kişiye okula nasıl geleceğini , hangi yollardan geçeceğini, hangi vasıtaları kullanacağı gibi anlatımlar tam olarak bir algoritma çalışmasıdır.

Algoritma, sorunumuzun çözümü için tasarladığımız yollar ve yöntemlerdir.

Uygun bir yol tarifi konabilir.

Algoritma ve Programlamaya Giriş

C) Programlama Dili Seçme Aşaması

Bu konuda öncelikle düşünülmesi gerekli olan bazı kriterler belirlenir. Bunlar, kısa ve/veya uzun süren bir çözüm olması, görsel tasarımın bulunup bulunmadığı, veritabanı kullanılıp kullanılmayacağı gibi kriterlerdir. Bu kriterlere bağlı olarak assembly, C, Visual Basic, Delphi gibi dillerden birine karar verilebilir.

Küçük bir kodlama resmi konabilir.

Algoritma ve Programlamaya Giriş

D) Kodlama Aşaması

Bu aşamaya kadar yapılan tüm çalışmalar bir çözüm ustaları tarafından yapılan çalışmalardır. Çözüm bulunmuştur ve hangi programlama dili ile kodlanacağı belli olmuştur. Ancak bu noktada da bir kodlama sanatından söz etmek gerekir. Çünkü programcılık bir sanattır. Aynı kodlamayı yapan iki programcının ürünleri arasında bir çok kriter incelendiğinde ciddi farklar görülecektir. Her ne kadar aynı sonuçları vermiş görünseler de arka planlarında farklar olacaktır.

Küçük bir kodlama resmi konabilir.

Algoritma ve Programlamaya Giriş

Kodlama Aşaması da tamamlandıktan sonra yapılan ilk işlem kodun yazım kurallarına uygun olup olmadığını anlaşılma işlemidir. (Syntax) adı verilen yazım kuralları sonuçları “hata mesajı” almayıncaya kadar devam ettirilir.

Hemen arkasından programın derlenmesi (compile) işlemi gerçekleştirilir.

Buraya kadar sadece programın çalışabilir bir dosya yaratması (executable file) sağlanmış olur.

Küçük bir kodlama resmi konabilir.

Algoritma ve Programlamaya Giriş

Derleme işlemi sadece sözdizimine ilişkin hataların yakalanabildiği bir işlemdir. Aslında programımızın doğru olarak çalıştığı, isteneni verdiği henüz belli değildir. Mantıksal hatalar, ya da analiz aşamasında kararlaştırılan işlemlere aykırı olan işlemler henüz temizlenmiş değildir.

Bu nedenle test işlemlerinin mutlaka yapılması gerekmektedir. Debug işlemi, programın mantıksal hatalarını ayıklayan bir işlemdir.

Küçük bir kodlama resmi konabilir.

Algoritma ve Programlamaya Giriş

Yazılan program hakkında son kullanıcılardan gelen yeni ve deęişiklik istekleri ile programın yeni versiyonları elde edilecektir. İşte böyle bir döngü içerisinde programcılık devam edecektir.

Programcıların yazdığı programların sayısı arttıkça bu konudaki ustalıkları da artacaktır.

Aslında ilk zorlukları aynen bir bisiklete ilk defa binerkenki zorluklarla benzeştirebiliriz. Bir kere bisikleti kullandıktan sonra nasıl, bir daha unutmazsanız , programlamada da kodlamayı başardıkça, ustalığınız artacaktır.

Küçük bir kodlama resmi konabilir.

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar başlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, önceki slaytları tekrar inceleyiniz.

Çözümlü Sorular

Programlama dillerinin hemen hepsinin de , bilgisayarın anlayacağı komutlara dönüştürülmesi gerekmektedir.

Bu işleme veyadenir.

Noktalı yerlere hangi ifade gelecektir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Bilgisayar programı yazılımında makro düzeyde 4 aşama bulunur.
Bunlar;

- A)..... Aşaması
- B)..... Aşaması
- C)..... Aşaması
- D)..... Aşaması

Noktalı yerlere hangi ifade gelecektir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

..... Aşaması Bilgisayar ortamında çözülecek olan sorunun, uygulama haline dönüştürülmesi için yapılan ve sorunun genel hatlarının ortaya çıkarıldığı aşamadır.

Noktalı yere hangi ifade gelecektir ?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

....., sorunumuzun çözümlü için tasarladığımız yollar ve yöntemlerdir.

Noktalı yere hangi ifade gelecektir ?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programlama Dili Seçme Aşamasında ele alınması gerekli kriterlerden birinin adını yazınız.

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

..... bir sanattır.

Noktalı yere ne gelecektir?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

....., programın mantıksal hatalarını ayıklayan bir işlemdir.

Noktalı yere ne gelecektir?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programcıların yazdığı sayısı arttıkça bu konudaki ustalıkları da artacaktır.

Noktalı yere ne gelecektir ?

** **

Yanıt gelecek derste verilecek.

Ders Materyali

Bir önceki derste Türkçe yayını önerdim. Bu derste ise çok iyi bir İngilizce yayın öneriyorum. Bu kitabın hem okuması ve anlaması çok kolay olmasa da algoritma çalışmaları için tam bir kaynak kitap. Kitabın künyesini ve linkini veriyorum.

Kitabın Künyesi: Algorithmics The Spirit of Computing David Harel with Yishai Feldman ADDISON WESLEY

Ders Sonu

Gelecek derslerde daha fazla bilgiyi ve daha ileri C programlarını yapmak üzere.

Hoşçakalın.

Başarılar.

Algoritma ve Programlamaya Giriş

Programlama Yapmak

1. Yarıyıl

Hafta-03

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse programlama yapma ile ilgili konulara değinilecek ve Algoritma sözcüğü ve kavramları hakkında temel bilgiler vermeye devam edilecektir.

Haydi başlıyoruz...

****Musa El Harizminin Resmi konacak****

Çözümlü Sorular

Bilgisayar programı yazılımında makro düzeyde 4 aşama bulunur.
Bunlar;

- A)..... Aşaması
- B)..... Aşaması
- C)..... Aşaması
- D)..... Aşaması

Noktalı yerlere hangi ifade gelecektir ?

**

Analiz Aşaması - Algoritma Aşaması - Programlama Dilinin
Seçilmesi - Kodlama Aşaması

**

Çözümlü Sorular

Programlama dillerinin hemen hepsinin de , bilgisayarın anlayacağı komutlara dönüştürülmesi gerekmektedir.

Bu işleme veyadenir.

Noktalı yerlere hangi ifade gelecektir ?

** Compile (derleme) - Interprinting (yorumlama) denir.

**

Çözümlü Sorular

..... Aşaması Bilgisayar ortamında çözülecek olan sorunun, uygulama haline dönüştürülmesi için yapılan ve sorunun genel hatlarının ortaya çıkarıldığı aşamadır.

Noktalı yere hangi ifade gelecektir ?

**** Analiz ****

Çözümlü Sorular

....., sorunumuzun çözümlü için tasarladığımız yollar ve yöntemlerdir.

Noktalı yere hangi ifade gelecektir ?

**** Algoritma ****

Çözümlü Sorular

Programlama Dili Seçme Aşamasında ele alınması gerekli kriterlerden birinin adını yazınız.

** görsel tasarımın bulunup bulunmadığı **

Çözümlü Sorular

..... bir sanattır.

Noktalı yere ne gelecektir?

** programcılık **

Çözümlü Sorular

....., programın mantıksal hatalarını ayıklayan bir işlemdir.

Noktalı yere ne gelecektir?

** Debug işlemi **

Çözümlü Sorular

Programcıların yazdığı sayısı arttıkça bu konudaki ustalıkları da artacaktır.

Noktalı yere ne gelecektir ?

** programların**

Algoritma ve Programlamaya Giriş

Geçen haftaki konularımızın içerisinde üzerinde en fazla durduğumuz aşamalar analiz ve algoritma aşamaları idi. Gerçekten de bir yerde çözümlenme sanatı adını da verebileceğim bu aşamalar , uygulamayı kodlama aşamasına kadar getiren ve etkinliğinin kalitesinin oldukça yüksek olması beklenen aşamalardır.

Hatta bu aşamalarda çözümlenme öncesi programcıların teknik alt yapılarını kuvvetlendirmesi konusunda da bir tavsiye niteliğinde uyarım olmuştu.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Takdir edersiniz ki, kullanıcı isteklerinin doğru ve etkin bir biçimde analiz edilmediği, dolayısı ile algoritmasının doğru belirlenmediği bir çözümün sakat doğacağı aşikardır. Algoritması yanlış kurulan bir program olacaktır.

Algoritma için daha önce verdiğim tanımı tekrar edersek, belirlenmiş bir sorunun çözümü için izlenecek adımların algoritmayı oluşturduğunu tekrarlamış oluruz. Çözüm için izlenecek adımların hiçbir karışıklığa meydan vermeden, anlaşılabilir bir şekilde metinsel veya şemasal biçimlerde oluşturulması bir algoritma çalışmasıdır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Algoritmanın, programlama dili bağımsız olduğunu biliyoruz. Bu ne demektir? Belirlenmiş herhangi bir sorunun bilgisayar ortamındaki çözümü / gerçekleşmesi için yapılan çalışma sırasında bir programlama diline özgü çözüm asla düşünülmez.

Çözüm algoritması gerçekleştirildikten sonra bunu en etkin uygulayacak programlama diline karar verilir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Algoritma çalışmaları bazı temel kriterleri içerirler. İşlem sırasının önceliklerinin çok net olması, işlemlerin sonlu olması, işlemlerin etkin olması, girdiler, çıktılar vb. gibi bilgiler ilk seferde söylenebilecek kriterlerdir.

Her algoritmada mutlaka tanımlanması gerekli değişkenler olacaktır. İşlemlerin ayrıntılı bir açıklanması gerekmektedir. Hangi veriler, hangi değişkenler tarafından temsil edileceklerdir?

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Algoritmalara örnek olarak, isterseniz buraya kadar anlattığımız program yazma işlemlerini verelim.

- 1.Başla
- 2.Tasarla ve Program Yaz
- 3.Program Komutlarını Bilgisayara Gir
- 4.Compile (Derleme) Yap
- 5.Derleme sonucu Hata Varsa
 3. Adıma Geri Dön
- 6.Bağlama İşlemini Yap
7. Bağlama İşleminde Hata Varsa
 3. Adıma Geri Dön
- 8.Programı Çalıştır ve Test Et
- 9.Hata Varsa 3. Adıma Geri Dön
- 10.Bitir

Bu algoritma anime edilecek.

Algoritma ve Programlamaya Giriş

Bir önceki slaytta verdiğimiz algoritmada eğer programı derlemeden önce çalıştırmayı yazmış olsaydık, bir mantık hatası yapmış olurduk.

Belki , test aşamasını ortadan kaldırıp, programın sadece çalışmasını sağlayacak bir değişiklik yapabiliriz.

Amacımız , soruna özgü, en mükemmel çözümü yakalamaktır.

Uygun bir resim konacak

Algoritma ve Programlamaya Giriş

Biraz daha sayısal bir algoritmaya ne dersiniz arkadaşlar? Ama bunu normal konuşma dilimizle gerçekleştirelim.

Kullanıcının tanımlayacağı birbirinden farklı 3 sayının çarpımını bulan bir algoritma yapmaya çalışalım.

- 1.Başla
- 2.Birinci Sayıyı Tanımla
- 3.İkinci Sayıyı Tanımla
- 4.Üçüncü Sayıyı Tanımla
- 5.Üç Sayıyı Birbirleri ile Çarp
- 6.Sonucu Ekranda Göster

Uygun bir resim konacak

Algoritma ve Programlamaya Giriş

Analiz sonrasında Algoritma çalışmalarının sunumu farklı biçimlerde olabilir. Buna algoritma sunum dilleri de diyebiliriz. Bunlar , günlük konuşma dili olabileceği gibi, akış şemaları, metinsel ifade ve programlama diline yakın olan sözde kod (Pseude code) şeklinde olabilirler.

Daha önce de vurguladığım gibi, işlem sıraları son derece önemlidir. Kullanılan ifadeler, anlaşılır, kısa ve öz olmalıdır.

Programcının bilgisayara iş yaptırdığı düşünüldüğü için komutlar emir şeklinde gösterilir.

Uygun bir resim konacak

Algoritma ve Programlamaya Giriş

Bir önceki algoritmayı bir de bu şekilde gerçekleştirelim.

- 1.başla
- 2.a sayısını tanımla
- 3.b sayısını tanımla
- 4.c sayısını tanımla
- 5.d = a * b * c;
- 6.d sayısını yaz.
- 7.dur.

Algoritmadaki başla ve dur ifadelerine dikkatinizi çekmek istiyorum. Mutlaka bir algoritmada bulunması gereken ifadelerdir.

Uygun bir resim konacak

Algoritma ve Programlamaya Giriş

Bilgisayar programları MİB (Merkezi İşlem Birimi) tarafından işletilir ve RAM(Random Access Memory) bellek içerisinde de tüm işlemlerin veri ilişkili sonuçları oluşturulur.

Bu durumu Mikro İşlemciler ve Assembler dersinizde 4. Yarıyıldan geniş olarak göreceksiniz.

Ancak tüm bu işlemlerin **bellekte birer sembol isimler altında tutulduğunu** ve programcıların bu **sembol isimler altında algoritmalarını yürüttüklerini** çok iyi anlamamız gerekir.

Bellekte **kendi isimleri altında , kendine ait veriyi tutan** bu sembol isimlere **DEĞİŞKEN** (Variable) denir.

Değişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

Tekrar edecek olursak, deęişkenler programcılarının daha programa ilk başladıklarında , üretmek için planladıkları **verilerin, sonuçların** her birini içlerinde tutacağı **anamlı birer sembolik isimlerdir.**

Deęişkenlere isim verme genel kuralları şunlardır:

- 1.Deęişken ismi alfabetik bir karakterle başlamalıdır.
(A – Z – a – z)
- 2.Deęişken ismi bir rakam (0-9) veya özel bir karakter (! , / , ? , * , vb.) ile başlayamaz.
- 3.Deęişken isimleri keyword (anahtar kelime) olamaz.
(Keyword : Program komutları)
- 4.Türkçe karakterler özel karakter sayılırlar.
küçük ' i ' kullanılabilir. Küçük ' ı ' kullanılmaz.

Deęişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

Haydi ilk defa bellekteki bir yere bir değer yerleştirelim.
“Bellekteki yer” dediğimizde bunu ne ile sağlıyorduk?

değişkenler

ile değil mi? Üstelik değişken adını verirken , yani **bellekte bir veri tutacak iken “bu veriye ilişkin anlamlı”** Bir isim vermiyor muyduk?

O zaman **“ogrenci_notu”** gibi anlamlı bir isme sahip bir değişken tanımlamış olalım.(arasındaki ‘_’ karakteri özel karakter sayılmamaktadır.)

Dikkat ederseniz **değişken isim yazma kurallarına** tam olarak uydum. Değil mi?

Bu arada **bellekte de artık bu isim altımda bir yerim var** değil mi?

Değişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

Bir öğrencimizin notu da 75 olsun. Bu bilgiyi ,

ogrenci_not = 75 ;

Atama deyimi ile, altında 75 not değeri, bellekte **ogrenci_not** ismi altında yerleşmiş oldu.

Belleğe yerleşen bu not değerini 1(bir) arttırmak istesek , **atama deyimimiz** şöyle olmalıdır :

ogrenci_not = ogrenci_not + 1 ;

Artık bellekte ogrenci_not değişkeni adı altında 76 not değeri var.

Değişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

Dikkat ederseniz ařağıdaki atama deyimi **matematik işlemleri olarak yanlıřtır.**

Ancak **programlama dillerinde doęru bir işlemdir.**

ogrenci_not = ogrenci_not + 1 ;

Eřitlięin **saę tarafındaki ifade oluşturulup**, daha sonra **eřitlięin sol tarafındaki deęişkene** (deęişkenin içerisinde önceden ne deęer olursa olsun temizlenerek)

ATANIR, KONUR, YERLEŐTİRİLİR.

Deęişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

Algoritma ve/veya programlama dilinde kullanılan aritmetik operatörler genel olarak şunlardır :

+ Toplama - Çıkarma / Bölme
*** Çarpma = Atama < Küçüktür**
<= Küçük Eşit > Büyüktür >= Büyük Eşit
== Eşit mi?

Değişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

Daha önceki yaptığımız algoritmayı yeniden ve son bilgilerimizi de kullanarak yazmaya çalışalım.

Ancak, kullanıcıdan bir bilgi istemeden belli sayı değerlerini kendimizin belirlediği bir durum yaratalım.

Örneğin , 1 den 10 kadar olan (10 Hariç) sayıların çarpımını bulan bir programın algoritmasını yazalım.

Bir sayaç değişkeni ve bir de çarpımların toplamını tutacağımız bir toplam değişkeni kullanmalıyız.

Sayaç değişkenininin 1 den başlaması ve 10 dan büyük olmamasını kontrol etmeliyiz. Toplam değişkenininin de her çarpım sonrası sonuçları kümülatif olarak toplamı biriktirmesini sağlamalıyız.

DEĞİL mi Arkadaşlar?

Değişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Algoritma ve Programlamaya Giriş

O zaman yazalım bakalım:

1.Başla

2.Toplam Değişkeni T, Sayaç Değişkeni i olsun

3.T = 1 ve i = 1 yapmalıyız (biriktirme ve kontrol)

4.T = T * i (çarpımların toplamı elde ediliyor)

5.i = i + 1 (sayaç değişkeni bir arttırılıyor)

6.i < 10 ise 4. Adıma git (sayaç kontrolü)

7.T değişkeni içerisindeki değeri yaz

8.Dur

Değişken kavramını açıklayacak bir animasyon veya bellek kutucukları gibi şekiller konabilir.

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar bařlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, nceki slaytları tekrar inceleyiniz.

Çözümlü Sorular

Çözüm için izlenecek adımların hiçbir karışıklığa meydan vermeden, anlaşılabilir bir şekilde metinsel veya şemasal biçimlerde oluşturulması bir çalışmasıdır.

Noktalı yerlere hangi ifade gelecektir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Algoritmanın, programlama dili bağımsız olduđu konusunu birkaç cümle ile anlatınız.

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Analiz sonrasında Algoritma çalışmalarının sunumu farklı biçimlerde olabilir. Buna algoritma sunum dilleri de diyebiliriz. Bunlar neler olabilirler ? Sayınız.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Ancak tüm bu işlemlerin **bellekte birer sembol isimler altında tutulduğunu** ve programcıların bu **sembol isimler altında algoritmalarını yürüttüklerini** çok iyi anlamamız gerekir.

Bu sembol isimlere programlama kavramı içerisinde ne ad verilir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Değişkenlere isim verme genel kurallarından 2 tanesini yazınız.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

İki sayının çıkarma işlemni yapan bir algoritma yazınız.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

ogrenci_not = ogrenci_not + 1 ;

Yukarıdaki atama işleminden önce ogrenci_not değişkeni içerisinde 49 not değeri varsa , atama deyiminden sonra hangi değere sahip olacaktır ?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

ogrenci_not = 49;

Yukarıdaki atama işleminden önce ogrenci_not değişkeni içerisinde 50 not değeri varsa , atama deyiminden sonra hangi değere sahip olacaktır ?

** **

Yanıt gelecek derste verilecek.

Ders Sonu

Gelecek derslerde daha fazla bilgi edinmek ve daha ileri C programlarını yapmak üzere.

Hoşçakalın.

Başarılar.

Algoritma ve Programlamaya Giriş

Programlama Yapmak

1. Yarıyıl

Hafta-04

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse programlama yapma ile ilgili konulara değinilecek ve Algoritma ve kavramları programlama hakkında temel bilgiler vermeye devam edilecektir.

Haydi başlıyoruz...

****Musa El Harizminin Resmi konacak****

Çözümlü Sorular

Çözüm için izlenecek adımların hiçbir karışıklığa meydan vermeden, anlaşılabilir bir şekilde metinsel veya şemasal biçimlerde oluşturulması bir çalışmasıdır.

Noktalı yerlere hangi ifade gelecektir ?

**

Algoritma

**

Çözümlü Sorular

Algoritmanın, programlama dili bağımsız olduğu konusunu birkaç cümle ile anlatınız.

** Algoritmanın, programlama dili bağımsız olduğunu biliyoruz. Bu ne demektir? Belirlenmiş herhangi bir sorunun bilgisayar ortamındaki çözümü / gerçekleşmesi için yapılan çalışma sırasında bir programlama diline özgü çözüm asla düşünülmez.

**

Çözümlü Sorular

Analiz sonrasında Algoritma çalışmalarının sunumu farklı biçimlerde olabilir. Buna algoritma sunum dilleri de diyebiliriz. Bunlar neler olabilirler ? Sayınız.

** günlük konuşma dili , akış şemaları, metinsel ifade , programlama diline yakın olan sözde kod (Pseude code) **

Çözümlü Sorular

Ancak tüm bu işlemlerin **bellekte birer sembol isimler altında tutulduğunu** ve programcıların bu **sembol isimler altında algoritmalarını yürüttüklerini** çok iyi anlamamız gerekir.

Bu sembol isimlere programlama kavramı içerisinde ne ad verilir ?

**** Değişken ****

Çözümlü Sorular

Değişkenlere isim verme genel kurallarından 2 tanesini yazınız.

** Değişken ismi alfabetik bir karakterle başlamalıdır.

(A – Z – a – z)

Değişken ismi bir rakam (0-9) veya özel bir karakter (! , / , ? , * , vb.) ile başlayamaz.

**

Çözümlü Sorular

İki sayının çıkarma işlemni yapan bir algoritma yazınız.

**

1. Başla
2. Birinci sayı a , ikinci sayı b , sonuç değişkeni s olsun
3. $a = 15$ olsun
4. $b = 9$ olsun
5. $s = a - b$
6. s nin içindeki sonuç değerini yazdır
7. Dur

**

Çözümlü Sorular

ogrenci_not = ogrenci_not + 1 ;

Yukarıdaki atama işleminden önce ogrenci_not değişkeni içerisinde 49 not değeri varsa , atama deyiminden sonra hangi değere sahip olacaktır ?

**** 50****

Çözümlü Sorular

ogrenci_not = 49;

Yukarıdaki atama işleminden önce ogrenci_not değişkeni içerisinde 50 not değeri varsa , atama deyiminden sonra hangi değere sahip olacaktır ?

**** 49****

Algoritma ve Programlamaya Giriş

Geçen haftaki en son yaptığımız algoritma çalışmasına tekrar bakarsak 6. adımdaki bir kontrole bağlı olarak , 4. adıma gidilme durumu dikkatinizi çekecektir.

Bu durum , programlama kavramları arasında en önemlilerinden biri olan **DÖNGÜ** kavramıdır.

1.Başla

2.Toplam Değişkeni T, Sayaç Değişkeni i olsun

3.T = 1 ve i = 1 yapmalıyız (biriktirme ve kontrol)

4.T = T * i (çarpımların toplamı elde ediliyor)

5.i = i + 1 (sayaç değişkeni bir arttırılıyor)

6.i < 10 ise 4. Adıma Git (sayaç kontrolü)

7.T değişkeni içerisindeki değeri yaz

8.Dur

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

DÖNGÜ kavramı , tekrar eden işlemleri belirli kontroller ve kendine özgü kuralları çerçevesinde otomatik hale getirerek gerçekleştiren bir programlama mantığıdır.

DÖNGÜ işlemlerinde bir karar verme meselesi vardır. Bu noktada bir mantıksal karşılaştırma yapılarak döngünün devam edip etmeyeceğine karar verilmiş olur.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Klavyeden iki dik kenarı girilen üçgenin hipotenüsünü bulan bir programın algoritmasını yazalım.

1.Başla

2.İlk dik kenar değerini a ya koy

3.İkinci dik kenar değerini b ye koy

4.Sonuç için c değişkenini tanımla

5. $c = (a * a + b * b) ^ { 1/2}$

6.c yi yazdır.

7.Dur

Uygun bir anime konulacak

Algoritma ve Programlamaya Giriş

Algoritmaların sunumları konusunda çok yaygın olarak kullanılan bir yöntem ise AKIŞ ŞEMALARI dır.

Belirlenmiş bir sorunun çözümü için izlenmesi gereken adımların , görsel olarak sembollerle ifade edilmiş şekline AKIŞ ŞEMASI denir.

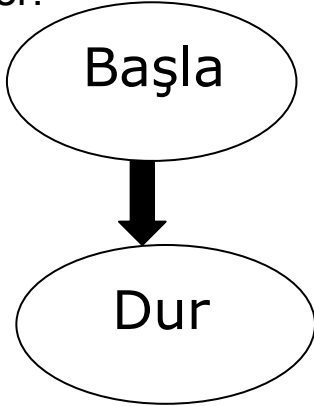
Akış şemalarının , diğer sunum şekillerinden farkı, adımların simgesel şeklindeki kutuların içine yazılmış olmasıdır. Ayrıca kutular arasındaki okların yönü ile de çözüme gidişin belirlenmiş olmasıdır.

Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Başla ve Dur Sembolü : Programın başlama ve bitme noktalarını belirler. İlgili yönlerle doğru da ok simgeleri ile gider.

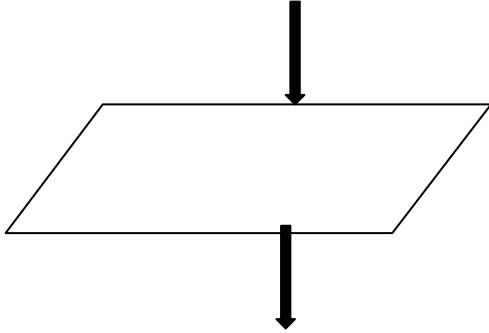


Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Girdi ve Çıktı Sembolü : Ekran, yazıcı, tarayıcı gibi bilgisayarın çevre birimlerine gönderilecek olan bilgileri gösterebilmek amacı ile kullanılır. Hangi birim için kullanılacaksa, o işlem şeklin içine yazılır.

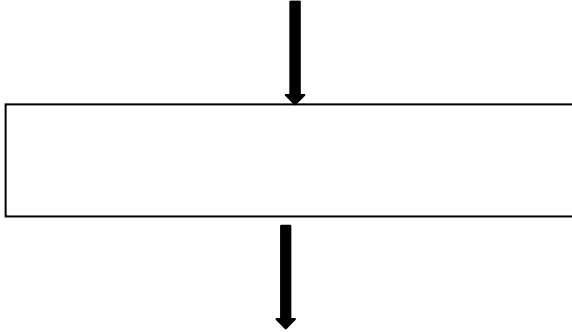


Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Atama / İşlem Sembolü: Atama işlemlerinin gösterildiği sembolik kutudur. $A = A + 5$; gibi ... veya $ogrenci_not = a * 2 + b * 3$; gibi işlemler, atamalar.

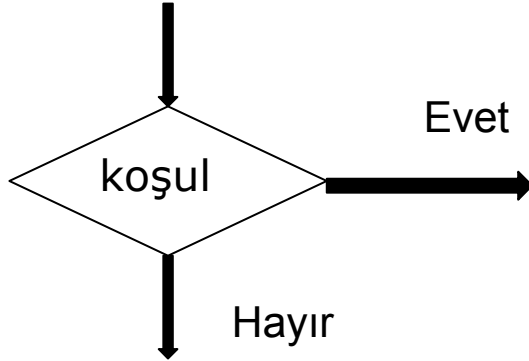


Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Karar Verme: Programın karar noktalarında kullanılan bir simgedir. "koşul" yazılan yere karşılaştırma işlemi yazılır. Sonuç, evet(true) ise bir yöne, hayır(false) ise diğer yöne akış devam eder.



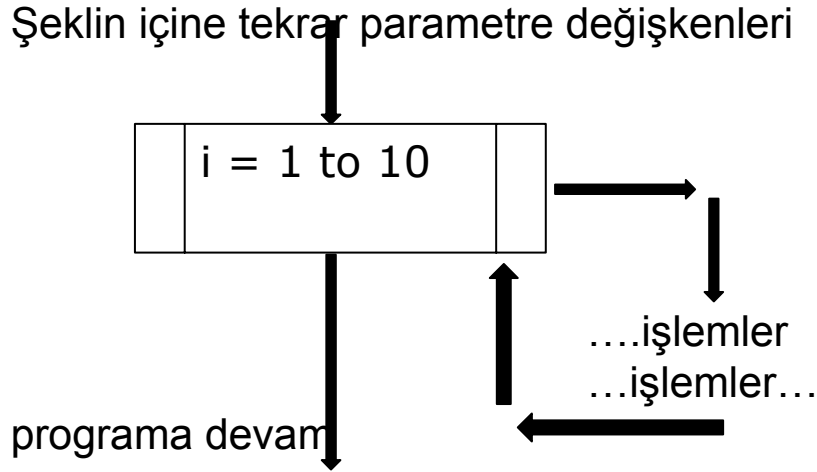
Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Döngü: Program akışı sırasında belirli blokların tekrarlanması gerektiğinde kullanılan bir simgedir.

Şeklin içine tekrar parametre değişkenleri girecektir.

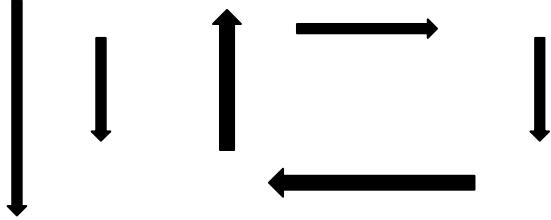


Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Akış Yönü: Algoritmanın akışını sağlayan çok önemli simgelerdir. Okların yönü mantıksal akışı sağlar.



Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Akış Şeması Görsel Kutuları ve Simgelediği İşlevler

Bağlantı: Karmaşık ve uzun bir görsellik gerektiren Algoritmalarda , programın akışının daha iyi anlaşılmasını sağlayan çok önemli simgelerdir. Bir bölümün , daha geride kalmış bir bölümle bağlantısını sağlamak üzere kullanılırlar.



Basit bir akış şeması konulacak

Algoritma ve Programlamaya Giriş

Basit bir algoritma yapalım. Daha sonra bu algoritmayı akış şeması olarak sunalım. Kullanıcı tarafından belirlenen bir sayının tek mi, çift mi olduğunu anlayıp, sonucu ekrana yazdıracak bir program olsun.

Sorunu analiz edersek, nelere gerek var diye ?

- 1.Kullanıcının tanımlayacağı bir sayı için bir değişken
- 2.Bu sayının tek mi, çift mi olduğunu anlayabilmek için karşılaştırma
- 3.Sonucu ekrana yazdırma

Lütfen yazılı algoritmayı şekle çevirmisiniz?

Algoritma ve Programlamaya Giriş

Algoritmayı yazalım :

- 1.Başla
- 2.Sayıyı oku a
- 3.(a % 2) eşit ise sıfıra
sayı çifttir yaz,
değil ise sayı tektir yaz.
3. Dur

Lütfen yazılı algoritmayı akış şemasına çevirmisiniz?

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar bařlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, 6nceki slaytları tekrar inceleyiniz.

Çözümlü Sorular

1. **Başla**
2. **Toplam Değişkeni T, Sayaç Değişkeni i olsun**
3. **T = 1 ve i = 1** yapmalıyız (biriktirme ve kontrol)
4. **T = T * i** (çarpımların toplamı elde ediliyor)
5. **i = i + 1** (sayaç değişkeni bir arttırılıyor)
6. **i < 10 ise 4. Adıma Git** (sayaç kontrolü)
7. **T değişkeni içerisindeki değeri yaz**
8. **Dur**

Yukarıdaki algoritmada kurulan döngü hangi adımlar arasında gidip gelmektedir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

DÖNGÜ kavramı , tekrar eden işlemleri belirli kontroller ve kendine özgü kuralları çerçevesinde otomatik hale getirerek gerçekleştiren bir programlama mantığıdır.

Doğru ise D , yanlış ise Y yazınız.

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

1. **Başla**
2. **İlk dik kenar değerini a ya koy**
3. **İkinci dik kenar değerini b ye koy**
4. **Sonuç için c değişkenini tanımla**
5. **$c = (a * a + b * b) ^ { 1/2}$**
6. **c yi yazdır.**
7. **Dur**

Yukarıdaki algoritmada kullanılan “değişken / değişkenler” hangisi / hangileridir ?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Akış şemalarının , diğer sunum şekillerinden farkı nedir ?
Sayınız .

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Akış Şeması Görsel Kutuları ve Simgelediği İşlevlerden
“ Girdi ve Çıktı Sembolü “ nün nasıl işlediğini anlatınız.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Akış Şeması Görsel Kutuları ve Simgelediği İşlevlerden “Karar Verme“ nin nasıl işlediğini anlatınız.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Kullanıcı tarafından belirlenen iki sayının hangisinin daha büyük olduğunu anlayan, büyükten küçük olanı çıkararak ekrana yazdıran bir çalışma yapınız.

- 1.Sorunu analiz ediniz, nelere gerek var diye ?
- 2.Algoritmayı yazınız
- 3.Algoritmayı akış şeması şeklinde gösteriniz.

** kadirmyo@gmail.com adresine

- 1.Adı Soyadı
- 2.Öğrenci No.su nu
yazarak gönderilecek

**

Yanıt gelecek derste sanal sınıfta verilecek.

Çözümlü Sorular

Akış Şeması Görsel Kutuları ve Simgelediği İşlevlerden “ Döngü” nün nasıl işlediğini anlatınız.

**

**

Yanıt gelecek derste verilecek.

Ders Sonu

Gelecek derslerde daha fazla bilgi edinmek ve daha ileri C programlarını yapmak üzere.

Hoşçakalın.

Başarılar.

Algoritma ve Programlamaya Giriş

Programlama Yapmak

1. Yarıyıl

Hafta-05

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse programlama yapma ile ilgili konulara değinilecek ve Algoritma ve kavramları programlama hakkında çalıřmalara devam edilecektir.

Haydi bařlıyoruz...

****Musa El Harizminin Resmi konacak****

Çözümlü Sorular

1. **Başla**
2. **Toplam Değişkeni T, Sayaç Değişkeni i olsun**
3. **T = 1 ve i = 1** yapmalıyız (biriktirme ve kontrol)
4. **T = T * i** (çarpımların toplamı elde ediliyor)
5. **i = i + 1** (sayaç değişkeni bir arttırılıyor)
6. **i < 10 ise 4. Adıma Git** (sayaç kontrolü)
7. **T değişkeni içerisindeki değeri yaz**
8. **Dur**

Yukarıdaki algoritmada kurulan döngü hangi adımlar arasında gidip gelmektedir ?

**

4 ve 6

**

Çözümlü Sorular

DÖNGÜ kavramı , tekrar eden işlemleri belirli kontroller ve kendine özgü kuralları çerçevesinde otomatik hale getirerek gerçekleştiren bir programlama mantığıdır.

Doğru ise D , yanlış ise Y yazınız.

** D **

Çözümlü Sorular

1. **Başla**
2. **İlk dik kenar değerini a ya koy**
3. **İkinci dik kenar değerini b ye koy**
4. **Sonuç için c değişkenini tanımla**
5. **$c = (a * a + b * b) ^ { 1/2}$**
6. **c yi yazdır.**
7. **Dur**

Yukarıdaki algoritmada kullanılan “değişken / değişkenler” hangisi / hangileridir ?

**** a , b , c ****

Çözümlü Sorular

Akış şemalarının , diğer sunum şekillerinden farkı nedir ?
Sayınız .

**

1. Adımların simgesel şeklindeki kutuların içine yazılmış olması.
2. Kutular arasındaki okların yönü ile de çözüme gidişin belirlenmiş olmasıdır **

Çözümlü Sorular

Akış Şeması Görsel Kutuları ve Simgelediği İşlevlerden
“ Girdi ve Çıktı Sembolü “ nün nasıl işlediğini anlatınız.

** Ekran, yazıcı, tarayıcı gibi bilgisayarın çevre birimlerine gönderilecek olan bilgileri gösterebilmek amacı ile kullanılır. Hangi birim için kullanılacaksa, o işlem şeklin içine yazılır..

**

Çözümlü Sorular

Akış Şeması Görsel Kutuları ve Simgelediği İşlevlerden “Karar Verme“ nin nasıl işlediğini anlatınız.

** Programın karar noktalarında kullanılan bir simgedir.”koşul” yazılan yere karşılaştırma işlemi yazılır. Sonuç, evet(true) ise bir yöne, hayır(false) ise diğer yöne akış devam eder.

**

Çözümlü Sorular

Kullanıcı tarafından belirlenen iki sayının hangisinin daha büyük olduğunu anlayan, büyükten küçük olanı çıkararak ekrana yazdıran bir çalışma yapınız.

- 1.Sorunu analiz ediniz, nelere gerek var diye ?
- 2.Algoritmayı yazınız
- 3.Algoritmayı akış şeması şeklinde gösteriniz.

** kadirmyo@gmail.com adresine

- 1.Adı Soyadı
- 2.Öğrenci No.su nu
yazarak gönderilecek

**

Çözümlü Sorular

Akış Şeması Görsel Kutuları ve Simgelediği İşlevlerden “ Döngü” nün nasıl işlediğini anlatınız.

** Program akışı sırasında belirli blokların tekrarlanması gerektiğinde kullanılan bir simgedir. Şeklin içine tekrar parametre değişkenleri girecektir.

**

Algoritma ve Programlamaya Giriş

Arkadaşlar, geçen dört hafta boyunca programcılık mesleğinin ilk ana temel bilgilerini sizlere aktarmaya çalıştım. Aslında analiz yapmak, algoritma , algoritma yazmak, algoritmayı farklı yöntemlerle sunmak gibi isimlendireceğimiz çalışmalarımız tamamen programlama dili bağımsız ÇÖZÜCÜ çalışmalardı.

Daha açık bir ifade ile, biz “kullanıcılardan gelen istekleri “ , bilgisayar ortamında gerçekleştirebilmek için gerekli bütün çalışmalarımızı yapmış sorunu çözmüş ve hangi adımlardan geçerek işimizi bitireceğimize karar vermiştik.

Ancak, bütün bu yaptıklarımızın bir aşaması daha kaldı. Şuna dikkatinize çekiyorum ki, bu çalışmalar yemeği pişirme noktasına kadar getiren çalışmalardı.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Ama yemeğimizi henüz pişirmedik. Bunun için çözücü çalışmalarımızı bilgisayarın anladığı bir programlama dilinde “KOD” lara dökmemiz gerekiyor.

Hangi programlama dili sorusunun , farklı kriterlere bağlı olarak cevabı bulunduktan sonra kod yazma aşamasına geçebiliriz.

Bizim sizlerle birlikte çalışacağımız programlama dili C dilidir. (si dili diye okunur) C dili yapısal programlama dili olarak 1980 li yıllarda bir devrim niteliğinde dünya üzerinde kullanılmaya başlamıştır. C diline aynı zamanda “ fonksiyonlar dili “ de denmesinin sebebi, en zor, karmaşık mantık kurgularına dahi etkili kolaylıklarla cevap verebilen , belli mantık gruplarını fonksiyon içinde iken dahi bloklara ayırabilen yeteneklere sahip yapısal bir programlama dilidir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

1978 yılında yaratıcıları olan, Dennis RITCHIE ve Brian KERNIGHAN ile ortaklaşa olarak “ The C Programming Language “ kitabını yazmışlardır. Kitap, bugün dahi güncelliğini yitirmeden yazılımcılara hizmet edebilecek kalite ve nitelikte olduğunu rahatlıkla söyleyebilirim. Kitabın yayınlanmasından sonra farklı işletim sistemlerinde de kullanılması ile birlikte çok yaygın olarak kullanılmaya başlamıştır.

Üçüncü kuşak dillerindendir. Ancak işletim sistemleri yazılımında kullanılması , nerede ise çoğu noktalarda assembly dili yerine dahi kullanılabilir olması , C dilinin daha da alçak seviyeli bir dil haline gelmesini sağlıyor.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Yazılması, kontrolü , kat kat daha kolay olmasına rağmen performans açısından etkili bir hıza sahip olması C dilinin halen günümüzde de son derece popüler olmasını sağlıyor.

Tabii bilgisayar teknolojilerinin her geçen gün hızla gelişmesi , mikro işlemcilerin gelişmesi, mimarilerin ucuzlaması ile birlikte bugün yaşantımızın içinde olan cep telefonları, uçak kontrol sistemleri, oyuncaklar, robotlar vb. gibi bir çok işlemcide C dili programlama yapılmıştır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

C dili ile programlama üstün performans sağlar, makinanın sahip olduğu tüm özellikleri bütünüyle kullanır. Analiz ve algoritma çalışmalarından sonra , son derece rahat ve kolay bir şekilde C dili programlama uygulanabilir. Algoritmik bir dildir.

Söz dizimi kuralları çok esnektir. Daha önce de vurguladığım gibi, bu esneklik, en karmaşık mantıkların dahi kolaylıkla programlanabilmesini sağlar.

Ancak, aynı esnekliğin diğer üçüncü kuşak dillerine göre daha zor kurallarının bulunmasına dikkat edilmelidir. Bu zor kurallar , aslında programcının tam kapasite ile çalışmasını sağlarlar.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

1970 li yıllardan beri üzerinde çalışılan, 1989 dan beri de kendine ait standartları oluşturulmuş bir dil olarak, diğer dillere nazaran taşınabilirlik özelliği çok fazladır.

Fonksiyonlar Dili olarak C dili, zengin hazır bir kütüphaneye de sahiptir. Ayrıca her programcının kendi kütüphanesini oluşturmasına ve kullanılmasına da izin verir. Alt yapısı çok güçlü bir dildir.

C programlama dilini kullanan programcıların dile çok hakim olmaları zaman içinde oluşacaktır. Ancak bu gelişim içerisinde , esnekliğin getirdiği , dilin inceliklerini kullanırken ki zorluklar, hataları debug ederken ki zorluklar C dilinin dezavantajları arasındadır.

Ama, “ GÜLÜ SEVEN DİKENİNE KATLANIR “

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Biraz da C++ programlama dilini tanıtmamız gerekiyor arkadaşlar. Çünkü, bizim derslerimizde kullanacağımız program geliştirme ortamı Visual Studio .NET olacaktır.

Bu geliştirme ortamında , Konsol Uygulamaları kategorisinde dünyanın programlama dilleri arasında yeri, en üst sırada bulunan C++ dili ile gerçekleştireceğiz.

C++ , C söz dizimini kullanan , C programlama dilinden bağımsız , nesne yönelimli programlamaya yönelik icat edilmiş bir programlama dilidir. C++ , NYP tekniğini içerebilmek için C nin geliştirilmiş ve eklentilerle güçlendirilmiş bir uygulama programıdır.

C++ , C dilini tamamen kapsamaktadır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Bir sonraki yılda da görsel programlama, internet programcılığı ve C# gibi alacağınız derslerin geliştirme ortamı .NET teknolojileri olacağı için, biz de bu dersten itibaren C++ dilini , C dili olarak kullanacağız bu ortamda. Böylece sizler geliştirme ortamına ilk yıldan itibaren ciddi bir alışkanlık kazanmış olacaksınız.

Ayrıca, C++ ile de ciddi bir tanışıklık kazanmış olacaksınız.

Bu durumu sizlere sanal sınıfımızda da daha detaylı açıklayacağım.

Sonraki 2 slaytımız rastgele seçtiğim bir C++ dili yazılmış programlama kodlarıdır. Bunları daha sonra çok iyi anlatabilir hale geleceksiniz.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Sadece C++ programlama kodları “neye benziyorlar” diye göstermek istedim.

```
// cppplus_clr_003.cpp : main project file.  
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
  
int main (void)  
{  
    char s1[50],s2[50];  
    int i;    cin.clear();  
    cout <<"Birinci Kelimeyi Girin:\n";  
    cin >> s1;    cout <<"ikinci Kelimeyi Girin:\n";  
    cin >> s2;
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

program devam ediyor....

```
for (i=0; (s1[i] == s2[i]) && s1[i] && s2[i]; i++);  
    if (s1[i] == s2[i])  
cout << "Girilmis olan s1 ve s2 aynidir.\n" << s1 << s2;  
    else if (s1[i] < s2[i])  
        cout << "s1 , s2 den once geliyor.\n" << s1 << s2;  
    else  
        cout << "s2, s1 den once geliyor.\n" << s2 << s1;  
        cin.get();  
return 0;  
}
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Program Kodlarını yazdığımızda, aslında biz “ **source code** “ ya da “ **kaynak kod** “ oluşturmuş oluyoruz. Yazdığımız kaynak kodun en küçük birimine atom (token) adı verilir.

Compile (derleme) işlemi bildiğiniz gibi yazdığımız source code u, object code a dönüştürür. Bunu yaparken de, algoritmanın mantığına göre “kaynak kod” u, en küçük birimlerine yani atomlarına ayırır ve bu arada söz dizimi hataları varsa onları bildirir.

Anahtar Sözcükler (keywords , reserved words)
Bunlar standart olarak programlama dili tarafından tanımlanmış anlamları olan sözcüklerdir. Anahtar sözcükler, bir değişken veya fonksiyon ismi olarak kullanılamazlar.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

auto, break, case, char, const, continue, default, do, double, else gibi sözcükler anahtar kelimelere verebileceğimiz örneklerden bazılarıdır.

C dilinde anahtar kelimelerin hepsi küçük harflerden oluşmuştur. C dilinin en önemli özelliklerinden biri de küçük büyük harflere karşı duyarlı olmasıdır. Dolayısı ile bir harfi bile yanlışlıkla büyük olan bir anahtar kelimeyi C derleyicisi algılayamayacaktır.

Programlama yaparken kullanacağımız veriler , sonuçlar, hesaplamalar yani “ farklı farklı türdeki değerler” in hepsi , RAM bellekte yer alabilmeleri için bir isme gereksinimleri vardır. Bu isimlere biz, değişken, fonksiyon ismi, makro adı vb. diyoruz.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

RAM Bellekten çok az bahsetmek istiyorum bu noktada arkadaşlar. CPU (Central Processing Unit) veya MİB (Merkezi İşlem Ünitesi) adını verdiğimiz Bilgisayarın en önemli biriminin program komutlarımızın icra edilmesi (uygulanması – gerçekleştirilmesi – çalıştırılması) sırasında değerlerin yerleştirildiği ve bilgilerin tümünün 0 ve 1 lerden oluştuğu bellek birimidir RAM.

İşte, değişken isimleri RAM bellekteki herhangi bir değeri temsil eder SİMGELER.

RAM Belleğin adreslenebildiğini yani her bayt ının adresinin belli olduğu bilgisini de kabaca size aktarırsam , değişken isimleri için şunu da söyleyebiliriz.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Değişken isimleri içlerinde barındırdıkları değerlerin RAM Bellekteki ADRESLERİDİR diyebiliriz.

Bu konuyu sanal sınıfta daha etkin bir şekilde konuşacağız.

İsim verme kurallarına uyularak verilen isimlerle değişkenler kullanılırlar.

Operatörler (İşleçler) ise aritmetik ve/veya mantıksal işlemleri gerçekleştiren Atomlardır. C dilinde 5 farklı operatör(işleç) gurubu , toplamda ise 45 tane operatör vardır.

a.**Aritmetik** , b. **Atama** , c. **Mantıksal**, d. **Bitsel**
e. **Adresleme** olmak üzere 5 farklı operatör gurupları vardır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Aritmetik İşleç Örnekleri :

+ , - , / , * , % , -- , ++

Atama İşleç Örnekleri :

= , *= , /= , += , -= , <<= , >>= , &= , ^= , |=

Mantıksal İşleç Örnekleri :

== , != , > , <> , <= , && , || , !

Bitsel İşleç Örnekleri :

& , ^ , << , >>

Adresleme İşleç Örnekleri :

sizeof() , & , * , ? , :

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Sabit deęerler, yani program ierisinde bařlangıta aldığı deęerlerinin hi deęiřmemesi istenilen isimlerdir.

String (karakter katarı) , C programlama dilinde, ift tırnak arasında tanımladığımız atomlardır.

Programlama sırasında özüm mantıklarımızı **deęiřkenler**, **operatörler** (iřleler) ve **sabitler** (deęiřmezler) üzerinde kurgularız. Bu oluřturduğumuz birleřime **İFADE** (Expression) denir.

Uygun bir resim konulacak

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar bařlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, 6nceki slaytları tekrar inceleyiniz.

Çözümlü Sorular

C diline aynı zamanda “ fonksiyonlar dili “ de denmesinin sebebi nedir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinin halen günümüzde de son derece popüler olmasının nedeni nedir?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinin dezavantajları nelerdir.

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Teknolojide C dili programlamanın yapıldığı yaşamımızda kullandığımız ürünlere örnekler veriniz.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C++ , nasıl bir programlama dilidir ?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programlama sırasında çözüm mantıklarımızı **değişkenler**, **operatörler** (işleçler) ve **sabitler** (değişmezler) üzerinde kurgularız. Bu oluşturduğumuz birleşime denir.

Noktalı yere ne gelecektir?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Program Kodlarını yazdığımızda, aslında biz
“ **source code** “ ya da “ **kaynak kod** “
oluşturmuş oluyoruz. Yazdığımız kaynak kodun en
küçük birimine adı verilir.

Noktalı yere ne gelecektir?

** **

Yanıt gelecek derste sanal sınıfta
verilecek.

Çözümlü Sorular

auto, break, case, char, const, continue, default, do, double, else gibi sözcükler verebileceğimiz örneklerden bazılarıdır.

Noktalı yere ne gelecektir ?

**

**

Yanıt gelecek derste verilecek.

Ders Sonu

Gelecek derslerde daha fazla bilgi edinmek ve daha ileri C programlarını yapmak üzere.

Hoşçakalın.

Başarılar.

Algoritma ve Programlamaya Giriş

Programlama Yapmak

1. Yarıyıl

Hafta-06

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse programlama yapma ile ilgili konulara değinilecek ve Algoritma ve kavramları programlama hakkında çalıřmalara devam edilecektir.

Haydi bařlıyoruz...

****Musa El Harizminin Resmi konacak****

Çözümlü Sorular

C diline aynı zamanda “ fonksiyonlar dili “ de denmesinin sebebi nedir ?

****En zor, karmaşık mantık kurgularına dahi etkili kolaylıklarla cevap verebilen , belli mantık guruplarını fonksiyon içinde iken dahi bloklara ayırabilen yeteneklere sahip yapısal bir programlama dili olmasdır..**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinin halen günümüzde de son derece popüler olmasının nedeni nedir?

** Yazılması, kontrolü , kat kat daha kolay olmasına rağmen performans açısından etkili bir hıza sahip olması **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinin dezavantajları nelerdir.

** C programlama dilini kullanan programcıların dile çok hakim olmaları zaman içinde oluşacaktır. Ancak bu gelişim içerisinde , esnekliğin getirdiği , dilin inceliklerini kullanırken ki zorluklar, hataları debug ederken ki zorluklar C dilinin dezavantajlarıdır.**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Teknolojide C dili programlamanın yapıldığı yaşamımızda kullandığımız ürünlere örnekler veriniz.

**

cep telefonları, uçak kontrol sistemleri, oyuncaklar, robotlar vb. gibi bir çok işlemcide C dili programlama yapılmıştır. **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C++ , nasıl bir programlama dilidir ?

** C++ , C söz dizimini kullanan , C programlama dilinden bağımsız , nesne yönelimli programlamaya yönelik icat edilmiş bir programlama dilidir. C++ , NYP tekniğini içerebilmek için C nin geliştirilmiş ve eklentilerle güçlendirilmiş bir uygulama programıdır.

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programlama sırasında çözüm mantıklarımızı **değişkenler**, **operatörler** (işleçler) ve **sabitler** (değişmezler) üzerinde kurgularız. Bu oluşturduğumuz birleşime denir.

Noktalı yere ne gelecektir?

**

İfade (Expression)

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Program Kodlarını yazdığımızda, aslında biz
“ **source code** “ ya da “ **kaynak kod** “
oluşturmuş oluyoruz. Yazdığımız kaynak kodun en
küçük birimine adı verilir.

Noktalı yere ne gelecektir?

** atom (token) **

Yanıt gelecek derste sanal sınıfta
verilecek.

Çözümlü Sorular

auto, break, case, char, const, continue, default, do, double, else gibi sözcükler verebileceğimiz örneklerden bazılarıdır.

Noktalı yere ne gelecektir ?

** anahtar kelimelere

**

Yanıt gelecek derste verilecek.

Algoritma ve Programlamaya Giriş

```
// kadir_c_plus_plus_02.cpp : main project file.
```

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
#include <string>
```

```
#include <sstream>
```

```
using namespace std;
```

```
int main (void)
```

```
{
```

```
    cout << " Bizim İlk Programımız " ;
```

```
}
```

Yukarıdaki kaynak kodu yazdık Visual Studio .NET IDE (Integrated Development Environment – Bütünleşik Geliştirme Ortamı) sinde çalıştırdık ve ekranda **" Bizim İlk Programımız "** görüntüsünü elde ettik.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Sizlere , Visual Studio .NET IDE (Integrated Development Environment – Bütünleşik Geliştirme Ortamı) ortamında hazırlanmış bir program kodunu göstermek istedim.

Tabii , bu programın her satırının ne anlama geldiğine dair sizlere tüm bilgilerimi aktaracağım. Ancak , .NET Teknolojilerinin çatısını oluşturan Framework denilen kütüphanesindeki hazır bileşenleri de sırası geldikçe tek tek ele alacağım.

Sizlere de .NET Ortamında bulunan MSDN adı verilen totarial bilgilerine sürekli olarak başvurmanızı ve orada aradığınız bir bilgiye ve ilgili örneklerine en hızlı erişebilmenin yollarını geliştirmenizi kuvvetle tavsiye ediyorum.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Sizlere , önceki hafta deęişken kavramı üzerinde gerekli bilgileri vermiş ve üzerinde kuvvetle durmuştuk. Ancak , türler üzerinde pek fazla konuşmamıştık.

Türü tanımlanmamış bir deęişkenin kullanılamayacağını söylersek sanırım türlerin önemini vurgulamış olurum.

C, C++ derleyicileri, tür bilgileri sayesinde deęişkeni bellekte nerede saklayacakları, veya deęişkenin programın çalışması sırasında hangi işlemlere tabi tutulacakları ile ilgili bilgileri öğrenirler.

Genel olarak C dilleri (C++ , C#) türe sıkıca baęlı dillerdir. Farklı türlerin birbirine dönüşümü olmadan kesinlikle kullanılamaz.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Önceden tanımlanmış veri türleri ile kullanıcı tanımlı veri türlerinden söz edebiliriz.

Önceden Tanımlanmış Veri Türleri :

Tamsayı(integer) :

int i, j, k ; (i, j, k programcının verdiği değişken isimleridir.)

short int sayi; // 2 bayt - 16 bit (-32768 ile +32767)

long int sonuc; // 4 bayt – 32 bit (-2147483648 ile 2147483647)

**unsigned short int maas; // 2 bayt 16 bit
// (0 ve 65,535)**

gibi tanımlamalarını örnek olarak verebiliriz.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Önceden Tanımlanmış Veri Türleri (devam)

Karakter(char) :

1 Bayt lık (8 bitlik) bellek bölümünde en soldaki bit (0/1) işaret (- veya +) biti olarak kullanıldığında -128 ile 127 değerleri arasında değişecektir.

İşaretsiz olarak ise 0 ile 255 arasındaki değerleri tutabilmektedir.

Örnek :

'A' karakteri ----> 65 ASCII kod karşılığıdır.

'a' karakteri ----> 97 ASCII kod karşılığıdır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Önceden Tanımlanmış Veri Türleri (devam)

Gerçel Sayı(float – double – long double):

C dilinde float, double ve long double olmak üzere 3 tane gerçel sayı veri türü bulunmaktadır ve üçü de işaretlidir.

Bu veri türlerinin bellekte yer alma özellikleri sistemden sisteme değişebilir, ancak sistemlerin büyük bir çoğunluğunda Institute of Electrical and Electronics Engineers (IEEE) standardına uyulmaktadır. Tüm sistemlerde float veri türü için 4 baytlık yer ayrılır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Önceden Tanımlanmış Veri Türleri (devam)

Gerçel Sayı(float – double – long double):

float 32 bit 4 bayt 6 basamak hassasiyet
1.17 x 10-38 ile 3.40 x 1038

double 64 bit 8 bayt 15 basamak hassasiyet
2.22 x 10-308 ile 1.170 x 10-38

long double 96 bit 10 bayt taşınabilir değil

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

C derleyicileri derlemeyi yapacakları mikroişlemcinin özelliklerine göre ,

A) 16 bitlik mikroşlemcilerde,

- char türü 1 byte
- int türü 2 byte(işlemcinin bir kelimesi kadar)
- short türü 2 byte(short = int)
- long türü 4 byte (long > int)

B) 32 bitlik mikroşlemcilerde,

- char türü 1 byte
- int türü 4 byte(işlemcinin bir kelimesi kadar)
- short türü 2 byte(short < int)
- long türü 4 byte (long > int)

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Bellek yerleşim özelliklerinden dolayı tamsayılarla yapılan işlemlerin, gerçel sayılarla yapılan işlemlerden çok daha hızlı gerçekleşeceğini söylemeliyiz.

Bu nedenle bir tam sayı türünün yeterli olması halinde, gerçel sayı türünün kullanılması programın okunabilirliğinin de azalmasına yol açabilir.

C dillerinde , değişken tanımlarını yapmadan , yani onların bellek bölümünde yerlerini almadan işlemlerde kullanılmaları hataya sebep olacaktır.

Bir değişkenin C derleyicisine tanıtılması türünün ve isminin derleyiciye bildirilmesi demektir. Bu bildirimden sonra derleyici , bu değişken için bellekte ne kadar yer ayıracağını ve bu ayrılan yere konan bilginin nasıl yorumlanacağını anlar.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Biraz da söylediklerimizi örneklerle pekiştirmeye çalışalım. C dilinin **küçük-büyük harf duyarlı** olduğunu ve anahtar kelimelerinin **küçük harfle** yazıldığını tekrar hatırlatayım.

Aşağıdaki iki tanımlama da aynıdır

char ch;

signed char ch ; // [signed] yazılması isteğe bağlıdır.

İşaretsiz Karakter Türü Tanımlama

unsigned char ch;

Aşağıdaki dört tanımlama da aynıdır

short sayi;

signed short sayi;

short int sayi;

signed short int sayi;

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

İşaretsiz short int Türü Tanımlama

unsigned short sayi;

Aşağıdaki üç tanımlama da aynıdır

int sayi;

signed int sayi;

signed sayi;

İşaretsiz integer Türü 2 adet aynı Tanımlama

unsigned int sayi;

unsigned sayi;

Aşağıdaki dört tanımlama da aynıdır

long sayi;

signed long sayi;

long int sayi;

signed long int sayi;

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

İşaretsiz long integer Türü 2 adet aynı Tanımlama

unsigned long sayi;

unsigned long int sayi;

Gerçek Sayılar İçin Tanımlama Türleri

float gercek_sayi;

double gercek_sayi;

long double gercek_sayi;

Programcılar , C dilinde altı kategoride isimlendirme yaparlar.Bunlar:

- 1.Değişkenler
- 2.Fonksiyonlar
- 3.Etiketler
- 4.Makrolar
- 5.Yapı ve Birlikler
- 6.enum Sabitleri

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

İsmlendirme kuralları ile ilgili bilgiler, 3. hafta ders notlarımızda verilmişti. İsterseniz , tekrar hatırlamak gerekiyorsa gidip onları inceleyin.

Aynı türe sahip farklı değişken isimlerini aralarına birer virgül koyarak tanımlayabilirsiniz.

Doğru Tanımlamalar:

```
char a1, a2, a3;
```

```
float f1, f2;
```

```
int maas, ucret;
```

Yanlış Tanımlamalar:

```
int sayi, float alan;
```

```
long int kare, short int sayi;
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

C dilinde blok dediğimizde { } bu parantezler arasındaki işlemlerden söz edilir.

Her blok sadece kendi içerisinde geçerli olacak olan değişken tanımlamalarına sahip olabilir.

Örnek:

```
.....  
{  
    int a, b;  
    a = 12;  
    b = a + 1;  
    {  
        int z;  
        z = a + b;  
    }  
}
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Bir önceki örneğimizi dikkatlice incelersek a, b değişkenlerinin kendi blokları içinde, z değişkeninin de kendi bloğu içerisinde tanımlanarak işlemlerde kullanılabildiklerini görebiliriz.

Burada herhangi bir hata yoktur. Önemli olan nokta şudur: z değişkeni, kendi bloğu dışında kesinlikle tanımsızdır. Kullanılırsa hata oluşacaktır.

“ ; “ Noktalı virgül atomu tek basına bir deyimdir. Her deyimin sonunda mutlaka bir “ ; “ olması gerekir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Değerler, değişkenlerin içinde yer alabildikleri gibi, doğrudan doğruya sabit değer olarak da programın içinde yer alabilirler.

Örnek:

```
x = x * 3.14 ; // 3.14 bir sabittir.
```

Değişkenlerin türleri olduğu gibi sabit değerlerin de türleri vardır. Derleyici , sabitlerin yazılış biçiminden de sabitleri saptayabilir.

-25 , 301, 35673, -45 (signed int sabitler)

23441L , 56785L (long int sabitler)

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

' a ' , ' ? ' karakter türü sabitler.

C dilinde karakter türü sabitler bellekte ASCII tablosundaki değerleri ile yer alırlar. Bu önemli özelliğe her zaman dikkat etmeniz gerekecektir.

Tekrar edecek olursak, küçük a karakterinin ASCII tablosundaki karşılığı 97 ise, bu karakter sayısal olarak kullanılmak istenirse kullanılabilir, karakter bilgisi olarak kullanılmak istenirse kullanılabilir.

Örnek:

char harf:

harf = ' A ' ; // aynı zamanda 65 sayısal değere sahip

harf = ' A ' + 1; // 66 değere yükselip B karakteri elde edilmiş olur.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Önceden tanımlanmış ters bölü karakter sabitleri
(escape sequences)

- '\0' Null Karakter
- '\a' Çan Sesi (Alert)
- '\b' Geri Boşluk (back space)
- '\t' Tab karakteri(tab)
- '\n' Aşağı Satır (new line)
- '\v' Düşey Tab(vertical tab)
- '\f' Sayfa İleri(form feed)
- '\r' Satır Başı(carriage return)
- '\"' Çift Tırnak(double quote)
- '\\' Ters Bölü(back slash)

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Programlamadaki deęişkenler ve sabitler ile yapılması gereken tüm işlemler **operatörler** ile gerçekleştirilir. Her bir operatör mikroişlemcinin bir işlem yapmasını sağlar ve sonuçlarında bir deęer oluşturmuş olurlar.

Örnek:

$maas = x * y / 5 + 8;$ deyimi $=, *, /, +$ operatörleri ile gerçekleştirmiştir.

Operatörler arasında işlem öncelikleri önemlidir. Öncelikler olmasaydı, operatörlerin neden oldukları sonuçlar mikroişlemciden mikroişlemciye deęişerek taşınabilirlik ciddi boyutta sorunlar yaratacaktı.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

C dilinde bulunan 45 operatör için 15 farklı öncelik seviyesi kullanılır. Aynı öncelik seviyesinde birden fazla operatör varsa , önceliğin sağdan sola mı, soldan sağa mı olacağı dikkate alınmalıdır. (associativity)

Konularımız ilerledikçe operatörleri daha yakından tanıyacağız. Şimdilik , önemli önceliklere ilişkin biraz daha bilgi verelim. Yeri geldikçe yine bahsedeceğiz.

() [] . ->	1. Seviye	soldan sağa
+ - ++ -- ! * & sizeof (tür)	2. Seviye	sağdan sola
* / %	3. Seviye	soldan sağa
+ -	4. Seviye	soldan sağa
<< >>	5. Seviye	soldan sağa
< > <= >=	6. Seviye	soldan sağa
== !=	7. Seviye	soldan sağa

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

C dilinde bulunan operatör dediğimizde terminoloji olarak operandlardan da söz etmek gerekir. Operand , operatörün işleme soktuğu bir ifadedir.

Tek operandlı operatörler: (unary operators)

İki operandlı operatörler: (binary operators)

Üç operandlı operatörler: (ternary operators)

Operandların neresinde kullanılmalıklarına göre :

Sonek (postfix) operatörler

Önek (prefix) operatörler

Araek (infix) operatörler

Neden Oldukları İşlere göre:

Aritmetik - İlişkisel - mantıksal - gösterici

Bitsel İşlem Yapan Operatörler

Özel Amaçlı Operatörler

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Operatörlerin Değer Üretmeleri:

$x = y + z$; işleminde $+$ operatörü , y ve z değişkenleri adı altındaki bellek bölümlerindeki değerleri toplayarak, x değişkeni için ayrılan bellek bölümüne toplam değeri koyacaktır. Bunu yapmadan önce x değişkeni için ayrılan bellek bölümü içerisinde inceden ne var ne yoksa temizlenerek, sadece toplam değerinin var olmasını sağlayan işlem yapılacaktır.

İki önemli durumu aktarmaya çalıştım arkadaşlar. Hem bir “atama” işlemini sizlere anlattım, hem de $+$ operatörünün nasıl değer üretmiş olduğunu.

Şöyle olsaydı, acaba hata olur mu idi?

$y + z$; C dili derleyicisi bu duruma sadece uyarı verir. Ancak baktığımızda üretilen bir değer yoktur.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Modulus Operatörü (%) :

İki operand alır . $x = 14 \% 3$; 14 ve 3 sabitlerini aldı.
x değişkeninde ne oldu ? 14 ün 3 e bölümünden kalanı
yani 2 değeri x değişkenine konur.

ornek:

$k = 27 - 6 * 4 + 7 / 2 - 14 \% 3$;

önce 6 ile 4 ü çarp 24, sonra 7 yi 2 ye böl 3, 14 ü 3 e
böl kalan 2 ,

$k = 27 - 24 + 3 - 2$;

$k = 4$; ve sonuç olarak k değişkeni için ayrılan bellek
bölümünde 4 değeri oluşmuş oluyor.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

```
void main (void)
{
    int k,m;
    k = 17;
    m = ++k;
    cout << k << m;
}
```

1. k ve m değişkenleri tamsayı türünde değer alacak şekilde tanıtılmış.
2. Bu tanıtım sonrası derleme sırasında kendilerine bellekte yerleri ayrılıyor.
3. k değişkeninin bellek bölümüne 17 değeri konuyor.
4. ++ operatörü k değişkeni öneki olduğundan ve ++k nın eş değer ifadesi de $k = k + 1$ olduğundan k nın değeri 18 oluyor.
5. m değişkeni için ayrılan bellek bölümüne 18 değeri konuyor. Ekrana 18 ve 18 değerleri k ve m için yazılıyor

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

```
void main (void)
{
    int k,m;
    k = 17;
    m = k++;
    cout << k << m;
}
```

1. k ve m değişkenleri tamsayı türünde değer alacak şekilde tanıtılmış.
2. Bu tanıtım sonrası derleme sırasında kendilerine bellekte yerleri ayrılıyor.
3. k değişkeninin bellek bölümüne 17 değeri konuyor.
4. m değişkeni için ayrılan bellek bölümüne 17 değeri konuyor
5. ++ operatörü k değişkeni soneki olduğundan ve k++'ın eş değer ifadesi de $k = k + 1$ olduğundan k'nin değeri 18 oluyor. Ancak Ekran 18 ve 17 değerleri k ve m için yazılıyor

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Karşılaştırma Operatörleri

< > <= >= == != küçükse, büyükse, küçük eşitse, büyük eşitse, eşitse, eşit değilse operatörleri çok sık kullanılan operatörlerdir.

Yine programcılık yolunda hızla ilerlerken etkin kontrol , karşılaştırma ifadeleri kurgulamanız gerektiğinde **mantıksal operatörler** çok fazla işinize yarayacaktır.

! değil (not) operatörü

&& ve operatörü (and)

|| veya operatörü (or)

örnek:

```
... if ( (k > m) && (k != 18) ) .....
```

Uygun bir resim konulacak

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar bařlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, 6nceki slaytları tekrar inceleyiniz.

Çözümlü Sorular

C, C++ derleyicileri, tür bilgileri sayesinde bellekte nerede saklayacakları, veya değişkenin programın çalışması sırasında hangi işlemlere tabi tutulacakları ile ilgili bilgileri öğrenirler.

Noktalı yere ne gelecektir?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Tamsayı (integer) türlerine örnekler veriniz.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Bir deęişkenin C derleyicisine tanıtılmasının ne için gerekli olduğunu, nelere etki ettiğini anlatınız.

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinin olduğunu ve anahtar kelimelerinin yazıldığını tekrar hatırlatayım.

Noktalı yerlere ne gelecektir?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programcılar , C dilinde altı kategoride isimlendirme yaparlar.Bunlar:

1.....

2.....

3.Etiketler

4.Makrolar

5.....

6.enum Sabitleri

Noktalı yerlere ne gelecektir?

** **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinde karakter türü sabitler bellekte ASCII tablosundaki ile yer alırlar. Bu önemli özelliğe her zaman dikkat etmeniz gerekecektir.

Noktalı yere ne gelecektir?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

$x = y + z$; işleminde $+$ operatörünün rolünü ve etkilerini anlatınız.

**

**

Yanıt gelecek derste sanal sınıfta verilecek.

Çözümlü Sorular

$$k = 56 - 6 * 7 + 9 / 3 - 17 \% 4;$$

işleminin sonucunda k değişkenine hangi değer atanacaktır?

**

**

Yanıt gelecek derste verilecek.

Ders Sonu

Gelecek derslerde daha fazla bilgi edinmek ve daha ileri C programlarını yapmak üzere.

Hoşçakalın.

Başarılar.

Algoritma ve Programlamaya Giriş

Programlama Yapmak

1. Yarıyıl

Hafta-07

Hazırlayan: Öğr.Gör. A.Kadir ERDEM

Giriş

Bu hafta, derse programlama yapma ile ilgili konulara değinilecek ve Algoritma ve kavramları programlama hakkında çalıřmalara devam edilecektir.

Haydi bařlıyoruz...

****Musa El Harizminin Resmi konacak****

Çözümlü Sorular

C, C++ derleyicileri, tür bilgileri sayesinde bellekte nerede saklayacakları, veya değişkenin programın çalışması sırasında hangi işlemlere tabi tutulacakları ile ilgili bilgileri öğrenirler.

Noktalı yere ne gelecektir?

** değişkeni

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Tamsayı (integer) türlerine örnekler veriniz.

```
** int i, j, k ; ( i, j, k programcının verdiği deęişken  
isimleridir.)
```

```
**
```

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Bir deęişkenin C derleyicisine tanıtılmasının ne için gerekli olduğunu, nelere etki ettiğini anlatınız.

** C dillerinde , deęişken tanımlarını yapmadan , yani onların bellek bölümünde yerlerini almadan işlemlerde kullanılmaları hataya sebep olacaktır.

Bir deęişkenin C derleyicisine tanıtılması türünün ve isminin derleyiciye bildirilmesi demektir. Bu bildirimden sonra derleyici , bu deęişken için bellekte ne kadar yer ayıracağını ve bu ayrılan yere konan bilginin nasıl yorumlanacağını anlar.

.**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinin olduğunu ve anahtar kelimelerinin yazıldığını tekrar hatırlatayım.

Noktalı yerlere ne gelecektir?

**

küçük-büyük harf duyarlı - küçük harfle **

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Programcılar , C dilinde altı kategoride isimlendirme yaparlar.Bunlar:

1.....

2.....

3.Etiketler

4.Makrolar

5.....

6.enum Sabitleri

Noktalı yerlere ne gelecektir?

** değişkenler
fonksiyonlar
yapı ve birlikler
**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

C dilinde karakter türü sabitler bellekte ASCII tablosundaki ile yer alırlar. Bu önemli özelliğe her zaman dikkat etmeniz gerekecektir.

Noktalı yere ne gelecektir?

**

değerleri

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

$x = y + z$; işleminde $+$ operatörünün rolünü ve etkilerini anlatınız.

** $x = y + z$; işleminde $+$ operatörü , y ve z değişkenleri adı altındaki bellek bölümlerindeki değerleri toplayarak, x değişkeni için ayrılan bellek bölümüne toplam değeri koyacaktır.

**

Yanıt gelecek derste sanal sınıfta verilecek.

Çözümlü Sorular

$$k = 56 - 6 * 7 + 9 / 3 - 17 \% 4;$$

işleminin sonucunda k değişkenine hangi değer atanacaktır?

**

16

**

Yanıt gelecek derste verilecek.

Algoritma ve Programlamaya Giriş

Değerli arkadaşlar, gerçek yaşam içerisindeki işlerimizi, akışlarımızı, sorunlarımızı , daha kontrollü, daha rahat, daha güvenli, daha hızlı bir biçimde bilgisayar yazılım teknolojileri ile yani programlar aracılığı ile gerçekleştirebiliriz.

Bu anlamda bilgilerimiz 6. hafta sonu itibari ile artık program yazma aşamasına doğru bizi getirmiş bulunmaktadır. Sizler, şu ana kadar orijinal bir sorunu ele alıp, akış şemaları ile algoritmalarını sunmayı, yani bir anlamda çözmeyi, daha sonra algoritmanızın C dilinde de program kodlarına dönüştürebilmek için değişkenlerinizi türleri ile birlikte saptayıp, onları tanıtmayı, onlarla birlikte operatörleri kullanarak soruna özgü değerlerin üretilmesini öğrendiniz.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Ancak her şeyi ile tamam olan bir program yazmak noktasına da henüz gelmediğinizi biliyorsunuz.

Bu noktaya kadar özellikle algoritmik çalışmalarda gösterdiğiniz performans küçümsenmeyecek bir şekilde program kodlarını kurgularken çok işinize yarayacaktır.

Lütfen , bu noktaya kadar eksikleri olduğunu düşünen arkadaşlarımla, hızlı bir şekilde bunları tamamlamalarını öneririm. Adım adım, sabırlı ve iradenizi koyarak yapacağınız çalışmalar sizi bu yolda mutlaka mutlu sona götürecektir. Ülkemizin yazılım sektörü, henüz çok daha yeni. Sizleri, yani işini iyi bilen programcıları bekliyor.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Programlamada Kontrol Deyimleri , aslında programcılık mesleğinin en can alıcı, sorunların çözümünde olmazsa olmaz aletler arasındadır. C dili bu anlamda çok zengin kontrol deyimlerine sahiptir.

IF Deyimi:

```
if ( koşul ifadesi )
    deyim1;
else
    deyim2;
```

if anahtar kelimesinden sonra aç ve kapa parantezler şarttır. Koşul ifadesi ne kadar kapsamlı veya ne kadar basit olursa olsun sonuçta oluşan iki adet değer vardır.

TRUE(doğru) veya FALSE(yanlış)

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

1- Eğer sonuç true ise, yapılacak işlem/işlemler hemen kapa parantez sonrası işlem yapılır. İşlem, bir deyimden daha fazla ise mutlaka { } parentezleri içine alınır.

2- Eğer sonuç true ve yapılacak bir işlem yoksa ; noktalı virgül ile sonlandırılır.

3- Bir if deyiminin koşul ifadesi sonucu “false” ise,
a. Ya “else” ile yapılacak işlemler { } parentezler arasına alınarak yapılır.
b. Ya da “else” hiç kullanılmadan program akışı kaldığı yerden devam eder.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Örnek :

```
int a;  
a = 12;  
if ( a > 10 ) a = 21;  
else a = 7;
```

Örnek:

```
int a;  
.....  
a = 12;  
if ( a > 10 ) a = 21;  
.....  
.....
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Parantezler içindeki “Koşul İfadesi” tüm aritmetik operatörler, mantıksal operatörler, atama deyimleri gibi mantık kurgulama (kontrol oluşturma) ları içerebilir.

Bu konuda, programcının ustalığı çok önemlidir. Sizlere bu konudaki önerim, önceleri son derece sade ve basit ve arka arkaya getirebildiğiniz kontrol aşamalarından geçmenizdir.

Karmaşık kontroller , başlangıçta sizi zorlayarak, gereksiz vakit kaybına ve emek kaybınıza yolaçabilir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

if deyimine sıra geldiğinde çalışması şöyle gerçekleşir:

koşul ifadesinin sayısal değeri hesaplanır. Bu değer, true ya da false olarak yorumlanır. Hesaplanan değer 0 (sıfır) ise false , 0 (sıfır) ın dışında herhangi bir değer ise true olarak yorumlanır.

Örnek:

```
int a = 12
```

```
if ( -5 ) ..... koşul sonucu true....
```

```
if ( 0 ) ..... koşul sonucu false....
```

```
if ( 115 ) ..... koşul sonucu true....
```

```
if ( a = 0 ) ..... koşul sonucu false....
```

```
if ( a > -5 ) ..... koşul sonucu true....
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Örnek:

```
char x = 'h';  
    if ( x >= 'a' || x <= 'z')  
        cout << "küçük karakter :" << x;  
    else  
        cout << "büyük karakter :" << x;
```

Burada , x in içindeki karakter değeri, karakter 'a' değerinden büyük veya eşit ise, VEYA , x in içindeki karakter değeri, karakter 'z' değerinden küçük veya eşit ise sonuç true dur. Değil ise false dur.

True durumunda () lardan sonra gelen işlem, false durumunda ise (eğer varsa) else deki işlem yapılır. else durumu belirtilmemişse , program kendi akışına devam eder.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Örnek:

```
if ( a * h / 2 ) {
```

```
    deyim1;
```

```
    deyim2;
```

```
    deyim3;
```

```
}
```

```
else {
```

```
    deyim1;
```

```
    deyim2;
```

```
    deyim3;
```

```
}
```

Bu örnekte de, true veya false oluşumlarında yapılacak işlem bir deyimden fazla olduğu için { } blok parantezlerimizi kullanarak sorunu çözmüş oluyoruz. Yani true durumunda true bloğu yürülmüş olacaktır.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

Örnek:

```
char x;  
if ( (x = getchar()) == 'k' ) {  
    deyim1;  
    deyim2;  
}  
else {  
    deyim1;  
    deyim2;  
}
```

Bu örnekte , ekrandan okunan ve x karakter değişkeninin içine alınan karakter değerinin 'h' karakter değerine eşit olup olmadığı koşulunun sonucunun true veya false olma hali elde edilmektedir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

if / else / if / else yapıları:

Arkadaşlar , buraya kadar if / else kontrolünün normal kullanım tekniklerini ve kurallarını öğrenmeye çalıştık. Şimdi ise sınırsız mantıksal çözümlere bizi götürebilecek bir olanağa doğru gidiyoruz.

Bu olanak , yani if else in bir ileri kullanımı şöyledir:

```
if (koşul1)
    if (koşul2) deyim1;
    else if (koşul3) deyim2;
        else if (koşul4) deyim3;
            else deyim4;
deyim5;
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

if / else / if / else yapıları:

Bir önceki slayttaki verdiğim örnek sizi korkutmasın arkadaşlar. Çünkü , hep aynı kural ile bu if / else yapılarını kullanıyoruz. Bu da nedir?

Bir if in else olmak zorunda değildir. Bir if in koşulunun true olması halinde () parantezlerinden sonraki işlem yapılır. False olması halinde ise varsa else deki işlem yapılır. Eğer else deki işlem yine bir if deyimini içeriyorsa aynı kurallar devam edip gider.

Sonuç da, bir else deyimi , kendisine en yakın if deyimine aittir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

if / else / if / else yapıları:

```
if ( k == 16) // yani k deęişkeni içindeki deęer , 16 ise  
    cout << " eőittir ";
```

ancak , yanlışlıkla karşılaştırmayı tek bir eşit karakteri ile yapsak, yani `if (k = 16)` yapsak, bu durumda önce bir atama işlemi gerçekleşecek, k deęişkenine 16 deęeri konacak, if açısından da 16 deęeri TRUE olduğundan (hatırlarsanız sıfırın dışındaki tüm eđerler TRUE, sıfır ise FALSE idi) program hiçbir hata mesajı üretmeden devam edip gidecektir.

`if ((k = fonksiyon()) != 0)` gibi bir if deyiminde fonksiyon görevini yapıp k ya deęer atayacak, daha sonra o deęerin sıfıra eşit olup olmadığı kontrol edilecek) Program herhangi bir hata mesajı vermeyecektir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

switch deyimi:

if / else yapılarından farklı olarak, bir ifadenin (veya bir değişkenin alabileceği) farklı değerlerinin her biri için farklı işlerin yapılabildiği bir olanaktır.

```
switch ( ifade )  
{  
    case ifade nin herhangi bir somut değeri : deyim1;  
                                                deyim2: break;  
    case ifade nin herhangi bir somut değeri : deyim1;  
                                                deyim2: break;  
  
    default : deyim1;  
              deyim2:  
  
}
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

switch deyimi:

Burada case anahtar sözcüğünden hemen sonra ve “iki nokta üstüste” karakterinden hemen önce gelen sabit değer , switch anahtar kelimesinden sonraki parantezler arasına yazdığımız ifadenin alabileceği bir değere eşit ise , ‘ : ‘ den sonra yapılacak işlemler bir veya birden fazla deyim olarak gerçekleştirilir.

Bu case girdikten sonra , bir sonraki case e girilmeyecek ise, break komutu ile switch bloğundan çıkılması gerekmektedir.

Eğer break komutu kullanılmaz ise, bir sonraki case işlemlerine otomatik olarak devam edilecektir.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

switch deyimi:

default anahtar kelimesi ise, switch bloğuna girildikten sonra , switch ifadesinin değeri, hiçbir case sabitleri ile eşleşmiyorsa, direkt olarak default da belirtilen işlemler gerçekleştirilir.

default anahtar kelimesi, bir switch bloğunda olmayabilir. Bu durumda , bloktaki case lerden hiç biri aktif olmuyosa, yani case lerdeki sabit değerlere, switch ifadesinin o sıradaki aldığı değer eşit olmuyorsa , bloktan hiçbir iş yapmadan çıkıp program akışı devam edecektir.

Ayrıca , default anahtar kelimesi switch bloğunun neresinde olursa olsun , aynı görevini yapacaktır. Case leri ve default un herhangi bir sıralama zorunluluğu yoktur.

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

switch deyimi:

```
if ( k == 43 ) {
    deyim1;
    deyim2;
}
else if ( k == 44 ) {
    deyim3;
    deyim4;
}
else if ( k == 45 ) {
    deyim5;
    deyim6;
}
else deyim7;

switch ( k ) {
    case 43 : deyim1;
              deyim2;
              break;
    case 44 : deyim3;
              deyim4;
              break;
    case 45 : deyim5;
              deyim6;
              break;
    default : deyim7;
}
}
```

Uygun bir resim konulacak

Algoritma ve Programlamaya Giriş

switch deyimi:

Arkadaşlar, bir önceki slaytı dikkati bir şekilde incelediğinizde, her iki yapının da kontrol mekanizmalarına nasıl uyarlanabileceğini, hangi esnekliklerinin, hangi algoritmik çözümlere karşılık gelebileceğini anlayacaksınız.

Analiz , çözümleme , algoritma ve kod aşamalarında her an elinizin altında olan ve etkin kullanmanız gereken bu yapılarla ilgili sanal sınıfta da bol bol uygulama yapacağız.

case anahtar kelime sabitlerin özellikleri :

case 23+45 : // bu doğru bir ifadedir.

case k+51: // yanlış bir ifade k değişkeni var.

case ' a ' : // doğru, sabit ifade

case 34.67 : // yanlış, tamsayı sabit değil

Uygun bir resim konulacak

Alıřtırmalar

Bir sonraki slaytta, bu hafta iřlenen konuyla ilgili alıřtırmalar bařlayacaktır.

Takıldıđınız yerler olursa, ilgili konu anlatımının olduđu, 6nceki slaytları tekrar inceleyiniz.

Çözümlü Sorular

if anahtar kelimesinden sonra aç ve kapa parantezler şarttır. Koşul ifadesi ne kadar kapsamlı veya ne kadar basit olursa olsun sonuçta oluşan hangi iki değer vardır?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Bir if deyiminin koşul ifadesi sonucu “false” ise,

- Ya “else” ile yapılacak işlemler { } parentezler arasına alınarak yapılır.
- Ya da “else” hiç kullanılmadan eder.

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

```
int a = 12;
```

```
if ( -55 ) ..... koşul sonucu .....
```

```
if ( 0 ) ..... koşul sonucu .....
```

```
if ( 115 - 100 - 3 * 5 ) ... koşul sonucu .....
```

```
if ( a = 0 ) ..... koşul sonucu .....
```

```
if ( a > -5 ) ..... koşul sonucu .....
```

Yukarıdaki if deyimlerinin sonuçları neler dir?

```
**
```

```
.**
```

Yanıt gelecek derste verilecek.

Çözümlü Sorular

if (koşul1)

if (koşul2) deyim1;

else if (koşul3) deyim2;

else if (koşul4) deyim3;

else deyim4;

deyim5;

İlk if deyimindeki koşul1 ifadesi false ise programda devam edilecek yer neresidir?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

```
if ( k == 0 )
```

```
    cout << " eşittir ";
```

yerine

```
if ( k = 0 )
```

```
    cout << " eşittir ";
```

yazılır ise program hata mesajı alır mı? almaz ise çalışma nasıl sonuçlanır ? Alır ise hangi hata mesajını alır ?

```
**
```

```
**
```

Yanıt gelecek derste verilecek.

Çözümlü Sorular

switch deyimi ,if / else yapılarından farklı olarak, bir ifadenin (veya bir değişkenin alabileceği) her biri için farklı işlerin yapılabildiği bir olanaktır.

Noktalı yere ne gelecektir?

**

**

Yanıt gelecek derste verilecek.

Çözümlü Sorular

Bu case girdikten sonra , bir sonraki case e girilmeyecek ise, ile switch bloğundan çıkılması gerekmektedir.

Eğer kullanılmaz ise, bir sonraki case işlemlerine otomatik olarak devam edilecektir.

**

**

Yanıt gelecek derste sanal sınıfta verilecek.

Çözümlü Sorular

```
switch ( k ) {  
    case 43 : deyim1; deyim2;  
    case 45 : deyim3; deyim4; break;  
    default : deyim5; }
```

Yukarıdaki switch deyiminin if karşılığını yazınız.

**

**

Yanıt gelecek derste verilecek.

Ders Sonu

Gelecek derslerde daha fazla bilgi edinmek ve daha ileri C programlarını yapmak üzere.

Hořakalın.

Başarılar.