



## Bilgisayar Programlama Öğretiminde Yapay Zekâ Tabanlı Bir Yazılım Sisteminin Geliştirilmesi ve Değerlendirilmesi

### Development of an Artificial Intelligence Based Software System on Teaching Computer Programming and Evaluation of the System

Aslıhan TÜFEKÇİ\*, Utku KÖSE\*\*

**ÖZ:** Bu çalışmanın amacı, üniversitelerde okutulmakta olan bilgisayar programlamaya dayalı derslere ilişkin öğretim kalitesini pozitif yönde artırmak amacıyla tasarlanmış ve geliştirilmiş olan yapay zekâ tabanlı bir yazılım sistemini tanıtmaktır. Bu bağlamda ilgili yazılım sistemi, 2011-2012 Güz Yarıyılı sırasında, Bilgisayar Programcılığı Ön Lisans programlarında verilmekte olan C Programlama – 1 derslerinde kullanılmış ve sistemin etkinliğini ve verimliliğini ortaya koymak adına, genel bir değerlendirme çalışması yapılmıştır. Bu bağlamda, deney ve kontrol gruplarını oluşturan toplam 120 öğrenci üzerinde bir deneysel değerlendirme çalışması uygulanmış ve tipik bir anket materyali aracılığıyla da öğrencilerin tecrübe edinilen sürece ve kullanılan yazılım sistemine karşı dönütleri dikkate alınmıştır. Elde edilen sonuçlar; geliştirilen bu yazılımın, genel anlamda bilgisayar programlamanın öğretiminde, spesifik anlamda ise özellikle C programlama dilinin öğretimi bağlamında, başarılı bir yaklaşım ortaya koyduğunu göstermektedir.

**Anahtar sözcükler:** bilgisayar programlama, yapay zekâ, bilgisayar eğitimi, bilgi tabanlı sistemler

**ABSTRACT:** Objective of this study is to introduce an Artificial Intelligence based software system, which was designed and developed in order to improve teaching quality in computer programming courses given at universities. In this sense, the related software has been used for C Programming – 1 courses given at Computer Programming programs, during the 2011-2012 Fall Term. In order to show effectiveness of the developed software system, a general evaluation study has been performed. In this sense, an experimental evaluation study has been applied on a total of 120 students forming experiment and control groups and student feedbacks for the experienced process and the used software system have also been considered via a typical survey material. Obtained results show that the system ensures a successful approach on teaching computer programming generally and C programming specifically.

**Keywords:** computer programming, artificial intelligence, computer education, knowledge-based systems

## 1. GİRİŞ

Bilgisayar öğretimi bağlamında, önem önceliği bakımından üst sıralarda yer alan konulardan birisi de, hiç kuşkusuz ki bilgisayar programlama yaklaşımının ve bu yaklaşıma bağlı ilgili bilgi ve becerilerin öğretimidir. Bu noktada, özellikle üniversitelerde yer alan, gerek bilgisayar tabanlı olan, gerekse faaliyet alanı bilgisayardan teknik açıdan faydalanmayı da gerektiren programların tümünde; öğrencilere bilgisayar programlamanın etkili ve verimli bir şekilde öğretimi, son derece önemli ve dikkate şayan bir yaklaşım olarak; akademik - bilimsel çalışma çabalarının aktif durumda kalmasına neden olmaktadır. Özellikle uluslararası ve ulusal kapsamdaki çok sayıda akademik çalışma, bu konu kapsamında oldukça kayda değerdir. Hatta son gelişmeler ışığında, bilgisayar programlama gibi kavramsal ve felsefi anlamda soyut bir yaklaşımın-becerinin, “somut öğretim materyalleri” yardımıyla (diğer bir deyişle “bilgisayar olmadan”), etkili bir şekilde öğretilmeye çalışıldığı görülmekte ve böylece bu durum da ilgili literatürün gelmiş olduğu noktayı da irdelemektedir (Wyeth, & Purchase 2000; Lawhead et al. 2002; Pollard, & Forbes 2003; Henderson 2008; Bilgi Çağı 2012). Bu noktada; Bilgisayar Bilimleri'nin önemli ve popüler bir kolu olan yapay zekâya dayalı yaklaşım, yöntem ve

\* Yrd. Doç. Dr., Gazi Üniversitesi, Gazi Eğitim Fakültesi, Ankara-Türkiye, e-posta: asli@gazi.edu.tr

\*\* Öğr. Gör., Uşak Üniversitesi, Bilgi İşlem Daire Başkanlığı, Uşak-Türkiye, e-posta: utku.kose@usak.edu.tr

tekniklerin de, bilgisayar programlama öğretimi aşamasında sıklıkla işe koşulması, dikkat çekici başka bir durumdur.

Yapay zekânın, özellikle karmaşık yapıdaki, teknik kavramların veya genel bağlamda ders konularının öğretimi ve öğrenilmesi aşamasında kullanılması, yaygın başvurulan bir yöntem olarak karşımıza çıkmaktadır. Konu daha detaylı irdelendiği zaman; günümüz literatüründeki eğilimin, problem çözümüne odaklanmanın yanında, problem kapsama alanına özen gösterilmesi ve sunulan çözümlerin, benzeri yöntemlere göre daha etkili olması adına çalışmalar yapılması yönünde de ilerlediğini ifade etmek mümkündür. Bu noktada değinilmesi gereken diğer bir hususa göre de; söz konusu eğilimin yapay zekâ kullanımının da ötesinde hayat bulması; özellikle öğrencilerin ihtiyaç duyduğu anlık, sarsılmaz uzman bilgilerine ve başarılı bir öğrenme süreci geçirilmesi adına önemli olan “dönüt” sürecinin istikrarlı bir şekilde yerine getirilmesine, zaten yapay zekâ tabanlı çözümlerin etkili bir biçimde yardımcı olmasıyla ilişkilidir. Kısacası, ilgili bağlamdaki çalışmalar, artık spesifik problemlerin öğretiminde yeni ya da daha etkili çözümler ortaya koyan; “akıllı” sistemlere-materyallere odaklanmış durumdadır.

Bahsedilen hedef alan kapsamında bu çalışmanın amacı, üniversitelerde okutulmakta olan bilgisayar programlamaya dayalı derslere ilişkin öğretim kalitesini pozitif yönde artırmak amacıyla tasarlanmış ve geliştirilmiş olan yapay zekâ tabanlı bir yazılım sistemini tanıtarak etkililiğini ve verimliliğini değerlendirmektir. Öne sürülen yazılım sistemi, yapay zekâ çalışmaları kapsamında önemli bir çalışma yaklaşımı olarak kabul gören, “bilgi-tabanlı sistemler” kategorisine girmektedir. Buna göre yazılım, sahip olduğu altyapıda kurulu durumda olan “uzman bilgisine dayalı bir şekilde”, C programlama diline ve genel anlamda bilgisayar programlamaya ilişkin problemlerin sağlanmasında ya da çözümünde, etkileşimli bir çözüm ortaya koymaktadır. Bu bağlamda ilgili yazılım sistemi, 2011-2012 Güz Yarıyılı sırasında, Bilgisayar Programcılığı Ön Lisans programlarında verilmekte olan C Programlama – 1 derslerinde kullanılmış ve sistemin etkililiğini ve verimliliğini ortaya koymak adına, genel bir değerlendirme çalışması yapılmıştır. Elde edilen sonuçlar; geliştirilen bu yazılımın, genel anlamda bilgisayar programlamanın öğretiminde, spesifik anlamda ise özellikle C programlama dilinin öğretimi bağlamında, başarılı bir yaklaşım ortaya koyduğunu göstermektedir.

## 2. KAVRAMSAL ÇERÇEVE

Bu çalışmada ortaya konan akıllı-yapay zekâ tabanlı yazılım sisteminin ve genel olarak gerçekleştirilen çalışmanın dayalı olduğu pedagojik nedenler birçok farklı açıdan değerlendirilebilmektedir. Okuyucuya daha rahat bir değerlendirme platformu ve detaylardan, optimum düzeyde arındırılmış bir bilimsel çalışma sunmak adına, söz konusu nedenler, maddeler halinde aşağıda listelenmiştir:

- Öncelikli olarak; öğrencilerin öğrenme sürecini daha verimli bir hale getirmek adına; görsel ve işitsel destekli, etkileşim düzeyi yüksek uygulamaların kullanılması, özellikle günümüz teknolojik gelişmelerinin ve bilgiye olan erişim ve kullanım imkânlarının dâhilinde, yaygın bir durum olarak ortaya çıkmaktadır (Low et al. 2003; Humar et al. 2005; Parvin 2010; Fan, & Geelan 2013; Palilonis 2013; Savas 2013; Su 2013; Yamaguchi et al. 2013). Bu çalışmaya konu olan yazılım sistemi de, mümkün olduğunca etkileşimli bir düzeyde olacak şekilde sunulmuştur.
- Daha önce de belirtildiği üzere; gerek bu çalışmanın, gerekse benzer nitelikte; karmaşık, teknik nitelikte kavramların ve ders konularının öğretiminde, etkili öğretim materyallerine olan ihtiyaç, belirgin bir şekilde ortaya çıkmaktadır (Dede et al. 1996; Dimitrova et al. 2001; Hu et al. 2012; Guo 2013). Bu noktada, bilgisayar programlama gibi soyutluk düzeyi olan, ilk aşamada karmaşık ve teknik düzeyi yüksek olarak değerlendirilebilen, önemli bir bilgi-beceri alanının etkili ve verimli bir şekilde öğretilmesi ve öğrenilmesi, yüksek öncelikli bir durum olarak değerlendirilmektedir. Genel anlamda bilgisayar destekli bir yazılımın, spesifik anlamda ise yapay zekâ

yaklaşımlarıyla etkinliği ve işlevselliği aşırı derecede artırılmış bir sistemin, başarılı sonuçlar yönünde pozitif bir ivme sağlayacağı, su götürmez bir gerçek olarak görülmektedir.

- Öğretim ve öğrenim süreçleri bağlamında nedenler üzerinde daha detaylı durduğumuz takdirde, öğrencilerin ihtiyaç duyduğu istikrarlı “dönüt” mekanizmasının ve sarsılmaz, tutarlı, yanılmaz nitelikte “uzman bilgisinin”, akıllı bir yazılım sistemiyle başarılı bir şekilde yerine getirilebileceği görülmektedir. Daha önce de bahsedilen “etkileşimli” yapının sağlanmasında büyük pay sahibi olacak olan dönüt mekanizmasının, özellikle öğrencilerin kendi öğrenme süreçlerini etkili bir şekilde yaşamalarına ve öğretilen konu ne olursa olsun, kendi bilgi düzeylerini oluşturabilmelerine çok büyük bir ölçüde katkı sağlayacağı, literatürde de önem verilen bir konu olarak karşımıza çıkmaktadır (Sales 1993; Clark, & Dwyer 1998; Foote 1999).
- Çalışmayla ilgili daha genel bir perspektiften olmakla birlikte, son derece önemli olan diğer bir husus da, gerek öğretim, gerekse öğrenim süreçlerinin yerine getirildiği platform ya da yaklaşımların niteliği olmaktadır. Buna göre; günümüzde yaygın bir kullanıma sahip olan ve yüz yüze eğitim süreçleriyle, uzaktan eğitim (e-öğrenme) süreçlerinin karışımıyla ortaya çıkan harmanlanmış öğrenme yaklaşımlarının yanında, öğrencilerin kendi başlarına etkili süreçler yaşamalarına da imkân tanıyan; çevrimiçi ve çevrimdışı uygulamaların rolünün de pedagojik bağlamda etkili olduğu görülmektedir (Akbiyık ve Seferoglu 2012; Çelen et al. 2013; Ersoy 2003; Deperlioğlu ve Köse 2010; Du et al. 2013; Weinstein et al. 2013). Örneğin; sunulan yazılım sisteminin çevrimiçi olarak, öğrencilerin istediği zaman ve mekânda ulaşabileceği nitelikte olması önemli bir noktadır. Bu konuyla paralel olarak, bu çalışma kapsamındaki yazılım sistemi de ilgili ihtiyaçları karşılayacak niteliktedir.

Çalışmanın dayalı olduğu pedagojik nedenler genel anlamda, kısa bir şekilde açıklandıktan sonra, öne sürülen yazılım sisteminin kullanım özellikleri ve işlevlerinin kısa bir şekilde ele alınması, değerlendirme öncesi bilgi sahibi olmak adına önemli bir kazanım olacaktır. Bu bağlamda bir sonraki bölüm ilgili yazılım sistemine adanmıştır.

### **3. BİLGİSAYAR PROGRAMLAMA ÖĞRETİMİNDE YAPAY ZEKÂ TABANLI BİR YAZILIM SİSTEMİNİN GELİŞTİRİLMESİ**

Daha önce de belirtildiği üzere; bu çalışmaya konu olan yapay zekâ-tabanlı yazılım sistemi, özellikle C programlama dili üzerinden bilgisayar programlamaya ilişkin bilgi ve becerileri kazandırmaya yönelik olarak ortaya çıkmaktadır. Bu noktada, programlama dili olarak C programlama dilinin seçilmesinin sebebi, günümüzün popüler programlama dillerinin (ister Web tabanlı, ister Form – Masaüstü tabanlı olsun) temelinde C programlama dilinin yatıyor olmasıdır. Zaten, yüz yüze eğitim süreci bağlamında, üniversitelerin müfredatında bu dersin yer almasının öncelikli sebeplerinden birisi de bu durumdur. Ayrıca, yine literatür incelendiği takdirde, C programlama dilinden farklı nitelikteki programlama dilleri için de; benzer özellikte ya da etkileşime dayalı çalışmaların gerçekleştirilmiş olduğu görülecektir (Corbett, & Anderson 1993; Corbett, & Anderson 1995; Brusilovsky, & Weber 1996; Corbett, & Bhatnagar 1997; Aedo et al. 2000; Jimenez-Diaz et al. 2005; Baghaei 2006; Galvez et al. 2009; Hu et al. 2012; Verdu et al. 2012; Guo 2013; Krishnamoorthy et al. 2013).

İlgili yazılım sistemi, daha önce gerek uzaktan eğitim yaklaşımıyla sunulan programlama tabanlı derslerde, gerekse diğer yüz yüze eğitim süreçlerinin yaşandığı, çeşitli ön lisans, lisans ve lisansüstü ders süreçlerine de konu olmuştur. Bu bağlamda daha önce yapılmış olan çalışmalar ve sonuçlar da yine ilgili literatüre kazandırılmıştır (Köse, & Deperlioğlu 2012). Söz konusu bu çalışmadan farklı olarak, bu makalenin kapsadığı çalışma yaklaşımının gerçekleştirildiği süreçte, ilgili yazılım sistemi daha genel bir “akıllı öğrenme ortamı” altyapısından çıkartılmış – tekil hale getirilmiş ve yine önceki duruma oranla, bilgi-tabanlı altyapısının kapasitesi, edinilen tecrübelerle

ve sonuçlara göre, neredeyse iki katına çıkartılmıştır. Bu durum, yazılım sisteminin “akıllı-karar verme mekanizmasını” daha esnek bir hale getirmiştir.

Genel bağlamda, yazılım sisteminin temel kullanım özellikleri, işlevleri ve bilgi-tabanlı altyapısı ile ilgili yaklaşımlar, takip eden alt-başlık altında olduğu gibi özetlenip, açıklanabilmektedir:

### 3.1. Yazılım Sisteminin Kullanım Özellikleri ve İşlevleri

Yapay zekâ tabanlı yazılım sisteminin arayüzü, öğrencinin sistemle etkileşime kolay bir şekilde girebilmesine olanak tanıyan, aynı zamanda hızlı, anlaşılır ve kolay kullanılabilir bir yapıda tasarlanmıştır. Buna göre, yazılım sistemini kullanırken “sürükle-bırak”, “sürükleyerek-seç”...vs. gibi işlevler yerine getirebilmektedir. Benzer şekilde sistemin klavye aracılığıyla daha hızlı bir kullanım tecrübesi sağlamak adına; kısayol-tuğu desteği de bulunmaktadır. Söz konusu yazılımdan genel bir ekran-görüntüsü, *Şekil 1* altında sunulmuştur.



Şekil 3.1. Yazılım sisteminden genel bir ekran-görüntüsü

Yazılım sisteminin, kolay kullanımını (hem öğrenciler, hem de öğretim elemanları için) sağlamak adına ön tanımlı olan özelliklerinden birisi de; hem öğrenciler hem de öğretim elemanları için sunulan Araç Çubuğu üzerindeki kontrollerdir. Söz konusu bu kontroller kullanılarak, arayüzde ekrana gelmiş olan problem tanımlarına uygun çözümleri içeren C programlama dili tabanlı programlar, öğrenciler tarafından kolaylıkla yazılabilmektedir. Yine bu problemlerin tanımları ve olası uygun çözüm yapıları da, öğretim elemanları tarafından kullanılabilen “Yazılım Yönetim Arayüzü” platformu üzerinden sisteme eklenmektedir. Şekil 1’de sunulan ekran görüntüsünden de anlaşılacağı üzere, tipik bir C programı yapısı, programın her farklı kesimini tanımlayan; renkli vurgulamalarla ekranda gösterilmektedir. Gösterimde tutulan bu kesimlerin düzenlenmesi, silinmesi ya da farklı bir konuma taşınması gibi işlemler, kolay kullanımı amaçlayan sürükle-bırak işlevleri ya da kısayol-tuşları aracılığıyla yerine getirilebilmektedir. Yazılım sisteminin etkileşimli özellikleri sayesinde öğrenciler, problem çözümlerine yönelik program kodlarını oluştururken çeşitli dönüt işlevleriyle yönlendirilmekte

ve bu mekanizma da kolay kullanımı en etkili biçimde sağlamaya yönelik, sürekli aktif bir durumda, arkaplanda çalışmayı sürdürmektedir.

Verilen problem için çözüm niteliğinde bir kod yapısı oluşturulduktan sonra, ilgili C programlama dili kodunun çalıştırılması, yine arayüzde sunulan “Derle” düğmesi yardımıyla gerçekleştirilmektedir. Düğmeye tıklandıktan sonra gerçekleşen görev, tipik bir C programlama dili derleyicisi ortamında gerçekleşen, program derleme ve çalıştırmaya benzer yapıda bir yazılım operasyonu olarak değerlendirilebilir.

İçerisinde hatalar içeren C programlama dili tabanlı bir çözüm, arayüzde çeşitli dönüt mekanizmaları yardımıyla öğrenciye bildirilmektedir. Bu noktada, programda yer alan fonksiyon yapıları ya da programlama diline özgü yazılan kod blokları, koyu renklerle gösterilmekte ve hata yapılan kod yapıları ve ilgili kod satırları arayüzde görsel kontrollerle vurgulanmaktadır. Hatalı yazılmış bir çözüm koduna ait; derleme işlemi sonrası gösterilen tipik bir yapı, Şekil 2’de gösterilmektedir.

Çözüm Kodu:

```
#include <stdio.h>
#include <conio.h>

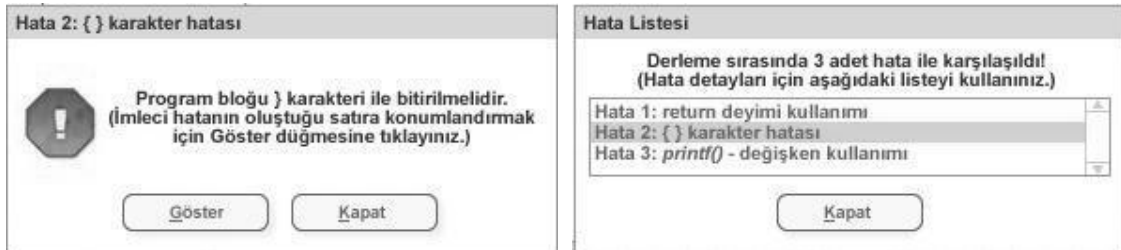
int toplama(int, int);

int main()
{
    int toplam,a,b;
    printf("İki sayı girin : ");
    scanf("%d %d",&a,&b);
    toplam = toplama(a,b);
    printf("%d ve %d nin toplamı", a,b,toplam);
}

int toplama( int x, int y )
{
    int sonuc;
    sonuc = x + y
    return sonuc;
}
```

Şekil 3.2. İçerisinde çeşitli hatalar içeren bir çözüm kodu [Koyu renkle vurgulanmış blokta hata(lar) bulunmaktadır]

Dönüt mekanizmasına sadık kalınarak, vurgulanan hatalara ilişkin çeşitli uyarı ve yönlendirme mesajları da, arayüzde ekrana getirilmektedir. Öğrenci, bu bildirim ve yönlendirme niteliğindeki mesajlarda sunulan kontrolleri kullanarak, hataların bulunduğu kod kesimlerine yönlenebilmekte veya çözüm odaklı yeni dönütler alabilmektedir. Şekil 3 altında, ekranda gösterilen bazı mesajlara örnekler sunulmuştur.



Şekil 3.3. Ekranda gösterilen dönüt mesajlarına örnekler

Gerçekleştirilen yönlendirmeler ve düzeltmeler ışığında, doğru çözümlerden birisini getiren öğrenciye, yeni problemler yazılım sistemi tarafından otomatik olarak sunulmakta ve böylece ilgili öğretim – öğrenim süreci sürdürülmektedir.

### 3.2. Yazılım Sisteminin Bilgi-tabanlı Altyapısı

Açıklaması yapılan problem ve çözüm odaklı etkinliklerin ve bu etkinliklerin etkili ve tutarlı bir şekilde değerlendirilebilmesinin arkasında yatan en önemli altyapı, daha önce de açıklanan “bilgi-tabanlı sistem” yapısı olmaktadır. Gerçekleştirilen bu çalışmada sunulan yazılım sisteminde bilgi-tabanlı yaklaşım, literatürde “Kısıt-tabanlı Modelleme (Constraint-based Modeling)” olarak da adlandırılan ve temelde olası çözümlerin bir araya getirildiği bir çözüm uzayında tutulan verilere [doğru çözüm(ler)e yönelik kısıt kümesine] bağlı olarak dönüt mekanizmasının işletilebildiği modelleme yöntemine bağlıdır (Galvez et al. 2009). Bu noktada Kısıt-tabanlı Modelleme, Ohlsson’un “hatalardan öğrenme” teoremine de dayanmaktadır (Ohlsson 1994).

İfade edilen çözüme yönelik kısıtların, belli bir alanda tutulması aşamasında, Ohlsson ve Rees tarafından öne sürülmüş olan bir notasyon yapısı kullanılmaktadır (Ohlsson, & Rees 1991). Notasyon yapısı ile ilgili genel kabuller şu şekildedir:

- Her bir kısıt  $\langle K_u, K_g \rangle$  çiftiyle ifade edilmektedir. Burada  $K_u$  uygunluk kısıtını,  $K_g$  ise gerçekleşme kısıtını ifade etmektedir.
- Tanımlanan her bir kısıt, genel anlamda problem ve çözüm uzayını tanımlamaktadır. Bu aşamada ilgili kısıtlar aynı zamanda koşul yapılarıyla [IF...then (Eğer...ise)] örgütlenmektedir (Örneğin; Eğer  $K_u$  gerçekleştiyse,  $K_g$  de gerçekleşmelidir; aksi takdirde “başka bir mekanizmayı” çalıştır.)

Söz konusu notasyona uygun olarak, C programlama diline ilişkin bilgi-tabanını oluşturabilecek bazı örnek kısıt tanımlamaları şu şekilde olmaktadır:

- $K_u$ ="bir atama operatörü var" ve  $K_g$ ="ilgili operatörün sağ tarafında geçerli yapıda bir ifade" olmalıdır.
- $K_u$ ="problem kullanıcı tanımlı değişken ister" ve  $K_g$ ="ilgili kod yapısında 'klavyeden okuma' işlevleri" olmalıdır.
- $K_u$ ="fonksiyonda 'return' işlevi kullanılmaz" ve  $K_g$ ="ilgili fonksiyon ismi başında 'void' ifadesi" olmalıdır.
- $K_u$ ="problem öz yinelemeli fonksiyon kullanımı ister" ve  $K_g$ ="ilgili kod return işlevleri var" olmalıdır.  
...vb.

Daha önce de değinildiği üzere; ilgili kısıtlarla oluşturulan kurallar, yazılım sisteminin altyapısında tutulmaktadır. Bu yolla, tutarlı ve yanılmaz uzman bilgisi işe koşulmak üzere, yazılım sisteminin ilgili algoritmik kod yapısına sunulmaktadır. “Bilgi-tabanını” oluşturmak adına tanımlanan toplam 732 kuralı, C programlama dili kapsamındaki problem ve çözüm odaklarına göre, *Tablo 1*'de belirtildiği üzere sınıflandırmak mümkündür.

**Tablo 1: Bilgi-tabanı İçerisindeki Kurallar**

Sınıf (Açıklama)	Kural Sayısı
Yazım (Yazım hatalarına karşı tanımlı dil bilgisi kuralları)	212
Fonksiyon (Fonksiyonların kullanımına yönelik her türlü kural)	131
Veri (Veri türleri ve kullanımına yönelik kurallar)	120
Hafıza ve Veri Modelleri (Etkin hafıza kullanımına ve veri modellerine yönelik kurallar)	42
Kütüphane (Yaygın kütüphane dosyalarının ve bağlı fonksiyonların kullanımına ilişkin her türlü kural)	135
Özel Çözümler (Bazı özel algoritmik çözümlere yönelik olarak tanımlanan kurallar)	71
Diğer (Yukarıdaki sınıflardan herhangi birisine girmeyen, spesifik ve tekil yapıdaki kurallar)	21
Toplam tanımlı kural sayısı	732

*Tablo 1*'den de anlaşılacağı üzere; bilgi-tabanı içerisinde yer alan kurallar, daha çok program yazımı, fonksiyon kullanımı, veri türleri ve kütüphane dosyalarının kullanımı yönünde toplanmaktadır. Bu noktada, bilgisayar programlarının yazımı aşamasında gerçekleşen hatalarının çoğunun özellikle yazım aşamalarında meydana gelmesi, en çok tanımlı kural sayısının yazım kategorisi altında olmasına neden olmuştur. Ayrıca ele alınan C programlama dilinin fonksiyon mantığı üzerine inşa edilmiş olması, benzer paralellikte kütüphane dosyaları üzerinden çalışan fonksiyonlara bağlı olarak şekillenen bir dil olması ve yine bu noktada, veri türlerinin problem çözümlerinde kritik roller oynaması, bağlı kategoriler altında da çok sayıda kural tanımlanmasına neden olmaktadır. Daha önce de ifade edildiği üzere; yazılım sisteminin önceki sürümünün yer aldığı çalışmadan elde edilen tecrübelerle geliştirilmiş olan bu bilgi-tabanı yapısı, yine C programlama dili ilgili olan diğer yazım ve kullanım şekillerine ilişkin, kritik kuralları da bünyesinde taşımaktadır.

Çalışmaya konu olan yazılım sistemi, kuşkusuz ki önceki başlıklar altında ifade edilen problemlere çözüm üretmek adına ortaya konan ve bu yolda özellikle yapay zekâ tabanlı bir yaklaşımdan destek alan bir yapıda, ilgili literatüre alternatif bir yaklaşım sunmaktadır. Bu noktada, ilgili yazılım sisteminin etkinlik ve verimliliğini ortaya koymak ve bu yolla; söz konusu çalışmanın, ifade edilen amaçlar doğrultusunda sonuçlara ulaşmış olduğunu anlamak adına, yazılım sisteminin genel bir değerlendirme sürecine tabi tutulması gerekmektedir. Bu nedenle; çalışmanın ilerleyen paragrafları, ifade edilen yönde şekillenmiştir.

## 4. YÖNTEM

Dayandığı temeller, kullanım özellikleri ve işlevleri ile yapay zekâ tabanlı altyapısı açıklanan yazılım sisteminin etkililiğini ve verimliliğini yorumlamak adına, çalışma niteliğine uygun çeşitli değerlendirme ve analiz süreçleri gerçekleştirilmiştir. Süreçlere dair detaylı bilgiler ilerleyen alt başlıklar altında sunulmaktadır.

### 4.1. Araştırma Modeli

Söz konusu yazılım sisteminin tutarlı bir şekilde değerlendirilmesi amacıyla, bilgisayar programlama yaklaşımının işe koşulduğu ortamlarda yazılım sisteminin aktif olarak kullanılması ve kullanım sonucunda elde edilen dönütleri – verileri analiz ederek birtakım sonuçlara ulaşılması önemlidir. Bu bağlamda; çalışma kapsamındaki yazılım sistemi bilgisayarlı programlama dersleri alan öğrencilerin kullanımına sunulmak suretiyle, etkinlik ve verimlilik yönünde bir araştırma çalışmasına geçilmiş ve bu yönde olmak üzere, niceliksel bir araştırma yaklaşımı tercih edilmiştir. Bu yaklaşım bağlamında ise öncelikli olarak deneysel araştırma modeli üzerinde durulmuş, ardından bir anket çalışması yardımıyla da yaklaşım desteklenmiştir. Veri toplama yönünde daha detaylı bilgi ilerleyen paragraflarda sunulduğu gibi; öncelikli olarak, araştırmaya konu olan çalışma grubunun da bilinmesi gerekmektedir.

### 4.2. Çalışma Grubu

Gerçekleştirilen değerlendirme süreci kapsamında, 2011-2012 Güz Yarıyılında verilmiş olan C Programlama – 1 dersinde yer alan toplam 120 öğrenci dikkate alınmıştır. Bu popülasyon üzerinde değerlendirme sürecine uygun olarak gerçekleştirilen değerlendirme ve analiz yaklaşımlarına uygun olarak çeşitli düzenlemeler gerçekleştirilmiştir. Gerçekleştirilen değerlendirme süreçleri ile ilgili olarak kullanılan veri toplama araçlarına değinmek, elde edilen verilerin ve analiz edimlerinin anlaşılması adına dikkate değerdir.

### 4.3. Veri Toplama Araçları

Veri toplama aşamasında; ifade edilen öğrencilerin öncelikli olarak deneysel gruplara ayrılması ve bu bağlamda yazılım sisteminin kullanımına yönelik bir değerlendirme yaklaşımı

izlenmesi yoluna gidilmiştir. Bu bağlamda yazarlar, C Programlama – 1 derslerine başladığı zamandan itibaren yazılım sistemini kullanan 60 öğrencinin yanında (deney grubu), yine 60 kişiden oluşan ve yazılım sistemini hiç kullanmayan bir grup (kontrol grubu) daha oluşturmuştur. İlgili gruplar (deney grubu ve kontrol grubu) oluşturulurken, öğrenci bilgi ve becerilerine uygun olarak, denk nitelikteki iki grubun oluşturulmasına özen gösterilmiştir. Gerçekleşen 14 haftalık süreç sonunda, öğrencilerin dönem sonu başarı notları ( $0,4 * vize + 0,6 * final$ ) genel olarak grupların başarı notlarını belirlemek amacıyla kullanılmıştır. Yazılımın etkinliğini görmek adına, ilgili yazılım sistemi Web platformu üzerinden sunulmuş ve böylelikle öğrencilerin istedikleri zaman ve mekânda, bilgisayar aracılığıyla öğrenme faaliyetlerine devam etmeleri amaçlanmıştır.

DeneySEL çalışma sonrasında gerçekleştirilen diğer bir veri toplama sürecinde ise, yazılım kullanmış olan 60 öğrencinin görüşlerini almak ve anlamak adına bir anket çalışması düzenlenmiş ve ilgili öğrencilere sunulmuştur. Söz konusu anket çalışması, Likert ölçeğine göre (1: tamamen katılıyorum; 2: katılıyorum; 3: kararsızım; 4: katılmıyorum; 5: tamamen katılmıyorum) ayarlanmış olan ve gerek yazılımın kullanımına, gerekse bilgisayar programlamanın öğrenimine ilişkin görüşleri ölçmeyi amaçlayan, toplam 10 maddeden oluşmuştur. Bu anket maddelerinin oluşturulması sürecinde uzman görüşünden yararlanılmıştır.

#### 4.4. Verilerin Analizi

Veri toplama araçları bağlamında, ders süreci kapsamında gerçekleştirilen deneysel değerlendirme yaklaşımıyla ilgili olarak; öğrencilerin genel anlamda C Programlama – 1 dersinde gösterdikleri başarı analiz sürecinde dikkate alınmıştır. Buna göre, bir önceki alt başlık altında da açıklandığı üzere, her öğrencinin yılsonu başarı notu, bağlı oldukları deneysel grubun başarılarını ortaya koyması adına analize dâhil edilmiştir. Bu analize göre, her bir gruptan ilgili dersi alan öğrencilerin oranı ve genel anlamda, 100'lük sistem üzerinden dönem sonu – ders ortalama başarı puanları da analiz sürecinde dikkate alınmıştır.

İfade edilen diğer veri toplama aracı olan anket çalışmasında ise; her bir maddeye karşı, Likert ölçeği kapsamında verilen dönütler analiz süreci dâhilinde dikkate alınmış ve bu paralellikte, çalışma kapsamında ortaya konulan yazılım sisteminin etkinliğini ve verimliliğini değerlendirme – yorumlama yolunda işe koşulan faktörlerden birisi olmuştur.

## 5. BULGULAR VE YORUMLAR

Veri toplama araçları ve açıklanan yöntemler üzerinden gerçekleştirilen değerlendirme süreci sonunda, deneysel olarak sürdürülen süreç neticesinde elde edilen bulgular ve yine söz konusu süreç sonrasında, deney grubuna mensup, 60 öğrenci tarafından verilen dönütler bağlamında ortaya çıkan bulgular yorumlanmak üzere analiz edilmiştir. Bu noktada öncelikli olarak, deneysel süreç sonunda elde edilen bulgular, *Tablo 2* altında gösterilmektedir.

**Tablo 2: Deneysel Değerlendirmeye İlişkin Bulgular**

Grup	Toplam Öğrenci Sayısı	Dersi Geçen Öğrenci Sayısı	Ortalama Puan (100 üzerinden)
Deney Grubu	60	52	81,56
Kontrol Grubu	60	32	53,78

*Tablo 2*'de gösterilen ve dönem sonunda elde edilen ilgili bulgular; çalışmaya konu olan yazılım sistemini kullanan öğrencilerin, akademik bağlamdaki başarı düzeylerinin arttığını göstermektedir. Bu noktada, özellikle C Programlama – 1 dersini geçen öğrenci sayısının, deney grubunda %52 oranlarına ulaşırken, kontrol grubunda %32 düzeylerinde kalması dikkat



çekicidir. Özellikle kontrol grubunda elde edilen oran, aynı dersin önceki dönemlerdeki öğrenciler kapsamında elde edilmiş oranlara yakın olması, yazılım sistemi olmaksızın, standart düzende işlenen dersin sonuçlarını tutarlı düzeyde görmek adına önemlidir.

Tablodan elde edilen diğer bir bulguya göre; hesaplama şekli daha önce ifade edilen ve vize ile final notları bağlamında elde edilen dönem sonu başarı puanları ortalaması, deney grubunda daha yüksek düzeylerde görülmektedir. Buna göre; söz konusu değer, yazılım sisteminin kullanan öğrencilerde 81,56 düzeyine ulaşması ve diğer yandan, kontrol grubunda ise 53,78 düzeyinde kalması, yazılım sisteminin akademik anlamda başarımı artırmış olduğunu anlamak adına son derece önemlidir.

**Tablo 3: Anket Çalışması ve Elde Edilen Bulgular**

Madde	Cevaplar				
	1	2	3	4	5
Sunulan yazılımla bilgisayar programlamayı daha etkili öğreniyorum.	38	13	9	0	0
Yazılım C programlama dilini iyi bir şekilde öğretmektedir.	39	9	2	0	0
Bu yazılım, yüz yüze öğretime göre daha etkili bir süreç sağladı.	26	22	10	2	0
Buna benzer bir faaliyette tekrar yer almak istemiyorum.	0	0	1	17	42
Ders başarı düzeyim, bu yazılımı kullandıktan sonra yükseldi.	36	19	5	0	0
Yazılım hızlı ve kolay kullanılabilir bir yapıda.	49	9	2	0	0
Yazılımı kullanmak, bana herhangi bir kazanım sağlamadı.	0	0	1	12	47
Yazılımın diğer programlama dilleri için de kullanılmasını isterim.	41	10	8	1	0
Yazılım bünyesinde sunulan problemler ve çözümleri beğendim.	38	10	8	3	1
Yazılıma Web’te, istediğim zaman ulaşabilmek, öğrenme sürecimi olumlu etkiledi.	45	10	4	1	0
Toplam katılımcı sayısı: 60					

Tablo 3’den de anlaşılacağı üzere, Bilgisayar Programcılığı öğrencileri, bilgisayar programlama (C programlama) öğretimi aşamasında, yapay zekâ temelli bir yazılım sisteminin kullanılmasından, pozitif yönde etkilenmişlerdir. Söz konusu anket maddeleri daha detaylı bir şekilde yorumlanacak olursa, şu sonuçlara ulaşıldığı ifade edilebilmektedir:

- Öğrenciler, yazılım sisteminin bilgisayar programlama yaklaşımını daha etkili öğrenme adına etken olduğunu düşünmektedir.
- Söz konusu yazılım sisteminin; bilgisayar programlama yaklaşımının öğretimi yanında, C programlama dilinin öğretilmesi yolunda da etkili olduğu düşünülmektedir.
- Öğrenciler, yazılımı kullanmanın yüz yüze öğretime göre daha etkili bir öğrenim süreci yaşattığını düşünmekte ve bu bağlamda benzeri bir süreçte yine yer almak istemektedir.
- Yine öğrenciler tarafından sağlanan dönütler, ders başarı düzeylerinin yazılım sisteminin kullanımı sonucunda arttığını ortaya koymaktadır.
- Yazılımın kullanımına ilişkin olarak; anket çalışmasında yer alan öğrenciler, ilgili yazılım sisteminin hızlı ve kolay kullanılabilir bir yapıda olduğunu düşünmektedir. Ayrıca yazılımın Web ortamında sunulması da öğrenciler tarafından oldukça olumlu bir şekilde karşılanmıştır.
- Öğrenciler, yazılım sistemi kapsamında sunulan problemleri ve ilgili çözümleri oldukça beğenmiştir.
- Söz konusu öğrenciler; aynı yazılım sisteminin, başka programlama dillerinin öğretimi – öğrenimi aşamasında da kullanılmasını istemektedir.

Gerek deneysel değerlendirme sürecinden, gerekse anket çalışması üzerinden elde edilen bulgular, geliştirilen yapay zekâ tabanlı yazılım sisteminin, genel anlamda bilgisayar programlama konusunda, özel anlamda ise C programlama dili konusunda, etkili ve verimli bir öğretim – öğrenim materyali sunduğunu göstermektedir. Elde edilen bu düzey, karmaşık ve teknik yapıdaki kavramların ve ders konularının öğretiminde, akıllı sistem yaklaşımlarının ne derece etkili olduğu konusunda, önemli fikirler vermektedir.

## 6. SONUÇ VE ÖNERİLER

Bu çalışmanın esas amacı, bilgisayar programlama bilgi ve becerilerinin kazandırılması ya da artırılması aşamasında yapay zekâ tabanlı yazılım sistemlerinden faydalanmanın ne düzeyde etkili bir yaklaşım olduğunu ortaya koymak ve bu kapsamda, Bilgisayar Programcılığı öğrencilerine yönelik bir bilimsel değerlendirme çalışması gerçekleştirmektir. Bu kapsamda incelendiği takdirde, çalışmaya konu olan “akıllı” yazılım sisteminin, daha önce de ifade edildiği üzere; bilgisayar programlama öğretiminde, kullanımı kolay, hızlı, etkileşim düzeyi yüksek, etkili ve verimli öğretim-öğrenim süreçlerinin yaşanabilmesine imkân tanıyan bir yaklaşım ortaya koyduğunu ifade etmek mümkün olmaktadır. Özellikle değerlendirme sürecinde, Ön Lisans Bilgisayar Programcılığı programlarında okumakta olan öğrencilere yönelik gerçekleştirilen değerlendirme çalışmalarından elde edilen bulgular, ilgili yazılım sisteminin olumlu yönde kazanımlar ortaya koyduğunu ve bu bağlamda da ilgili çalışmanın amaçlarına ulaştığını göstermektedir.

Çalışmadan elde edilen önemli bir çıkarım olarak; özel anlamda bilgisayar programlama dili öğretiminde takip edilebilecek etkili yöntemlerden birisinin de, bilgisayar tabanlı akıllı yazılım sistemlerinin kullanımı olduğu ifade edilebilmektedir. Bu kapsamda, literatürde yer alan çok sayıda çalışma, ilgili çıkarımı destekler nitelikte olduğu gibi, yerine getirilen bu çalışma da ilgili literatüre, paralel anlamda bir katkı sağlamış olmaktadır. Literatürde gerçekleşen bu çalışmaların, akıllı sistemlerin akademik öğretim sürecinde kullanılması aşamasında birçok yeniliği ve gelişimi de beraberinde getireceği de önemli bir öngörüdür.

## 7. KAYNAKLAR

- Akbıyık, C. ve Seferoglu, S. S. (2012). İlköğretim Bilişim Teknolojileri dersinin işleniş: Öğretmen görüş ve uygulamaları. *Kuram ve Uygulamada Eğitim Bilimleri (KUYEB)*, 12(1), 405-424.
- Aedo, I., Diaz, P., Fernandez, C., Martin, G. M., & Berlanga, A. (2000). Assessing the utility of an interactive electronic book for learning the PASCAL programming language. *IEEE Transactions on Education*, 43, 403-413.
- Baghaei, N. (2006). A collaborative constraint-based adaptive system for learning object-oriented analysis and design using UML. In V. P. Wade, H. Ashman ve B. Smyth (Eds.), *Adaptive hypermedia and adaptive web-based systems* (pp. 398-403). Berlin: Springer.
- Bilgi Çağı. (11.08.2012). *Bilgisayar olmadan, somut objelerle programlama dilini öğreten sistem*. [Çevrim-içi: [http://www.bilgicagi.com/Yazilar/10742-bilgisayar\\_olmadan\\_somut\\_objelerle\\_programlama\\_dilini\\_ogreten\\_sistem.aspx](http://www.bilgicagi.com/Yazilar/10742-bilgisayar_olmadan_somut_objelerle_programlama_dilini_ogreten_sistem.aspx)], Erişim tarihi: 26.09.2012.
- Brusilovsky, P., & Weber, G. (1996). Collaborative example selection in an intelligent example-based programming environment. *Proceedings of the International Conference on Learning Sciences*, 357-362.
- Clark, K., & Dwyer, F. M. (1998). Effects of different types of computer-assisted feedback strategies on achievement and response confidence. *International Journal of Instructional Media*, 25, 55-63.
- Corbett, A. T., & Anderson, J. R. (1993). Student modeling in an intelligent programming tutor. In E. Lemut, B. du Boulay ve G. Dettori (Eds.), *Cognitive models and intelligent environments for learning programming* (pp. 135-144). New York: Springer-Verlag.
- Corbett, A. T., Anderson, J. R., & O'Brien, A. T. (1995). Student modeling in the ACT programming tutor. In P. Nichols, S. Chipman ve B. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 19-41). New Jersey: Erlbaum.
- Corbett, A. T., & Bhatnagar, A. (1997). Student modeling in the act programming tutor: Adjusting a procedural learning model with declarative knowledge, *Proceedings of the International Conference on User Modeling*, 243-254.
- Çelen, F. K., Çelik, A., & Seferoglu, S. S. (2013). Analysis of teachers 'approaches to distance education. *Procedia - Social and Behavioral Sciences*, 83, 388-392.
- Dede, C., Salzman, M. C., & Bowen Lotfin, R. (1996). ScienceSpace: Virtual realities for learning complex and abstract scientific concepts, *Proceedings of the Virtual Reality Annual International Symposium*, 246-252, 271.
- Deperlioğlu, Ö. ve Köse, U. (2010). Web 2.0 teknolojilerinin eğitim üzerindeki etkileri ve örnek bir öğrenme yaşantısı, *Akademik Bilişim 2010 (AB 2010)*, Muğla, Türkiye.

- Dimitrova, V., Self, J., & Brna, P. (2001). Applying interactive open learner models to learning technical terminology. *Lecture Notes in Computer Science*, 2109, 148-157.
- Du, Z., Fu, X., Zhao, C., Liu, Q., & Liu, T. (2013). Interactive and collaborative e-learning platform with integrated social software and learning management system. *Lecture Notes in Electrical Engineering*, 212, 11-18.
- Ersoy, H. (2003). *Blending online instruction with traditional instruction in the programming language course: A case study*. Master dissertation, Middle East Technical University, Turkey.
- Fan, X., & Geelan, D. (2013). Enhancing students' scientific literacy in science education using interactive simulations: A critical literature review. *Journal of Computers in Mathematics and Science Teaching*, 32, 125-171.
- Foote, C. (1999). Attribution feedback in the elementary classroom. *Journal of Research in Childhood Education*, 13, 155-166.
- Galvez, J., Guzman, E., & Conejo, R. (2009). A blended e-learning experience in a course of object oriented programming Fundamentals. *Knowledge-Based Systems*, 22, 279-286.
- Guo, P. (2013). Online python tutor: embeddable web-based program visualization for cs education, *Proceedings of the ACM Technical Symposium on Computer Science Education*.
- Henderson, P. (2008). Computer Science unplugged. *Journal of Computing Sciences in Colleges*, 23, 168.
- Hu, M., Winikoff, M., & Cranefield, S. (2012). Teaching novice programming using goals and plans in a visual notation, *Proceedings of the Fourteenth Australasian Computing Education Conference*, 43-52.
- Humar, I., Sinigoj, A. R., Bester, J., & Hagler, M. O. (2005). Integrated component web-based interactive learning systems for engineering. *IEEE Transactions on Education*, 48, 664-675.
- Jimenez-Diaz, G., Gomez-Albarran, M., Gomez-Martin, M. A., & Gonzalez-Calero, P. A. (2005). ViRPlay: playing roles to understand dynamic behavior, *Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts*.
- Köse, U., & Deperlioglu, Ö. (2012). Intelligent learning environments within blended learning for ensuring effective c programming course. *International Journal of Artificial Intelligence and Applications*, 3, 105-124.
- Krishnamoorthy, V., Appasamy, B., & Scaffidi, C. (2013). Using intelligent tutors to teach students how APIs are used for software engineering in practice. *IEEE Transactions on Education*, Early Access Article.
- Lawhead, P. B., Duncan, M. E., Bland, C. G., Goldweber, M., Schep, M., Barnes, D. J., & Hollingsworth, R. G. (2002). A road map for teaching introductory programming using LEGO© mindstorms robots, *Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education*, 191-201.
- Low, A. L. Y., Low, K. L. T., & Koo, V. C. (2003). Multimedia learning systems: A future interactive educational tool. *The Internet and Higher Education*, 6, 25-40.
- Ohlsson, S., & Rees, E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition and Instruction*, 8, 103-179.
- Ohlsson, S. (1994). Constraint-based student modeling. In J. E. Greer ve G. I. McCalla (Eds.), *Student modeling: The key to individualized knowledge-based instruction* (pp. 167-189). Berlin: Springer-Verlag.
- Palilonis, J., Butler, D., & Pamela, L. F. (2013). Design interactive: A nonlinear, multimedia approach to teaching introduction to visual communication and principles of design. *Journal of Educational Multimedia and Hypermedia*, 22, 63-79.
- Parvin, R., Haque, N., Ahmed, N., Ahmad, R., Ali, I., Ara, R. et al. (2010). Is audio visual method better than traditional for medical students? - A better survey report. *Bangladesh Journal of Medicine*, 21, 60-64.
- Pollard, S., & Forbes, J. (2003). Hands-on labs without computers, *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*, 296-300.
- Sales, G. C. (1993). Adapted and adaptive feedback in technology-based instruction. In J. V. Dempsey, G. C. Sales (Eds.), *Interactive instruction and feedback* (pp. , 159-176). New Jersey: Educational Technology Publications.
- Savas, P. (2013). Thematic blogs: Tools to increase interaction, collaboration and autonomy among pre-service foreign language teachers. *Journal of Interactive Learning Research*, 24, 191-209.
- Su, M. (2013). Research of flash interactive features in courseware design. *Lecture Notes in Electrical Engineering*, 205, 537-545.
- Verdu, E., Regueras, L. M., Verdu, M. J., Leal, J. P., de Castro, J. P., & Queiros, R. (2012). A distributed system for learning programming on-line. *Computers & Education*, 58, 1-10.
- Weinstein, S., Pollack, K., Shibley, I., Williams, V. S., Wu, S.-W., & Ching, Y.-H. (2013). The blended learning initiative: A university-wide program to develop, deliver, and assess blended courses. In E. J. Francois (Eds.), *Transcultural blended learning and teaching in postsecondary education* (pp. 260-282). Hershey: IGI Global.

Wyeth, P., & Purchase, H. C. (2000). Programming without a computer: a new interface for children under eight, *Proceedings of the Australasian User Interface Conference*, 141-148.

Yamaguchi, R., Sakoi, A., & Hirakawa, M. (2013). Multimedia framework for application of spatial auditory information to learning materials. *Lecture Notes in Computer Science*, 7813, 161-166.

### Extended Abstract

Objective of this study is to introduce an Artificial Intelligence based software system, which was designed and developed in order to improve teaching quality in computer programming courses given at universities. In this sense; originating point of the study is difficulties in teaching the computer programming approach. All of the related difficulties are mostly based on intelligibility levels of course subjects (provided within a typical computer programming course) and intelligibility levels in this context can be adjusted to be suitable for students via effective and efficient learning approaches and also using feedbacks. At this point, it can be expressed that Artificial Intelligence based software systems are one of the most effective learning tools that can be used to reduce difficulties and provide more effective learning experiences for students.

It is also important that students learning the computer programming approach for the first time, can also find it difficult to understand “algorithmic thinking” and “principles of computer programming”. In this sense, the computer programming approach is connected with many abstract and theoretical subjects needing more advanced studies to be understood and learned. Especially in typical programming language courses, students often find it difficult to use their knowledge for designing and developing computer programs. Another important factor triggering the mentioned problem is complexity of today’s programming languages.

It must also be expressed that the main objective of Artificial Intelligence based software systems is improving students’ learning process via more advanced educational techniques. At this point, working principles of Artificial Intelligence based software systems are based on different pedagogical approaches and theories that have been developed in the education field. In this sense, “feedback” is some of the most important approaches that Artificial Intelligence based software systems are mostly built on. Nowadays, the feedback factor is widely used in many educational – instructional activities for providing more effective and efficient learning processes for students. Especially, it is a necessary approach for the courses, which are based on mostly technical and practical subjects. It is clear that students can learn abstract concepts and perform practical exercises better when they receive immediate feedbacks. Otherwise, both teachers and students may come across many difficulties during the education process.

In the context of the mentioned problems and subjects, lots of different scientific-research studies have been made to provide effective and efficient solutions for the mentioned problems in teaching computer programming. In addition to the theoretical approaches, there also exist some Artificial Intelligence based software systems that have been designed and developed to teach programming concepts in different programming languages. At this point, the software introduced in this study is a some kind of Artificial Intelligence based software system in which students can take some exercises by using an easy to use interface and teachers can create new C Programming based exercises by using the management interface provided in the system. For each exercise, the teacher can define the problem text and develop what would be the correct solution to that problem in the same way as a student would do. At this point, evaluation mechanism of the software system is based on a domain prepared according to the expert knowledge and domain expert knowledge of the software system can also be adjusted for specific exercises by using the management interface of the system. In more detail, the software system uses a student modeling called “Constraint Based Modeling” to handle the wide spectrum of solutions provided for problems of the C programming language.

Constraint Based Modeling is based on Ohlsson’s theory of learning from errors and proposes that the student can learn from the feedback generated as the result of an error. According to the technique, the domain includes some basic principles, which should be supported by all solutions for a given problem. A constraint based model represents knowledge about a domain as a set of “constraints” on correct solutions. The constraints select, out of the universe of all possible solutions, the set of correct solutions. As a result, they partition the universe of possible solutions into the correct and the incorrect ones and in this way, they provide some kind of approach for categorizing the problems and potential solutions.

The related software system introduced in this study has been used for C Programming – 1 courses given at Computer Programming programs, during the 2011-2012 Fall Term. In order to show effectiveness of the developed software system, a general evaluation study has been performed. In this sense, an experimental evaluation study has been applied on a total of 120 students forming experiment and control groups and student feedbacks for the experienced process and the used software system have also been considered via a typical survey material. According to the experimental evaluation study results; 52 of 60 experiment group students (who have used the software system) have passed the C Programming – 1 course with the average end-term success point of 81,56 whereas 32 of 60 control group students (who have not used the software system) have passed the related course with the average end-term success point of 53,78. On the other hand, student feedbacks received within the applied survey have uncovered a positive trend towards the used software system and the regarding educational process. Eventually; obtained results show that the system ensures a successful approach on teaching computer programming generally and C programming specifically.

---

**Kaynakça Bilgisi:**

Tüfekçi, A. ve Köse, U. (2013). Bilgisayar programlama öğretiminde yapay zekâ tabanlı bir yazılım sisteminin geliştirilmesi ve değerlendirilmesi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi [Hacettepe University Journal of Education]*, 28(2), 469-481.

**Citation Information:**

Tüfekçi, A., & Köse, U. (2013). Development of an artificial intelligence based software system on teaching computer programming and evaluation of the system [in Turkish]. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi [Hacettepe University Journal of Education]*, 28(2), 469-481.