

Öğrenci Numarası : \_\_\_\_\_ Adı Soyadı : \_\_\_\_\_

Sınav çoktan seçmeli ve klasik olmak üzere iki kısımdan oluşmaktadır. Çoktan seçmeli kısmın cevapları optik okuyucuya uygun cevap kağıtları üzerine işaretlenecektir. İlk **30dk** sonunda optik cevap kağıtları toplanacaktır. Kalan klasik soruları bu süreden sonra cevaplayabilirsiniz.

Grubunuzu işaretlemeyi unutmayınız!

Soru	26	27	28	29	Toplam
Puan	10	10	15	15	50
Not					

1. \_\_\_\_ tanımlamalar belirli bir blok ile ilişkilidir.(\_\_\_\_ declarations are associated with a specific block.)  
A. Global B. Belirli(Specific) C. Genel(General) **D. Yerel(Local)**
2. Bağlamdan bağımsız gramerler \_\_\_\_ adı verilen ve tanımlanan tüm öbeğin yerine geçen en üst seviye bir nonterminal içerir.(A context-free grammar has a special nonterminal called the \_\_\_\_ symbol which stands for the entire top-level phrase being defined.)  
A. ileri sembolü(forward) **B. başlangıç sembolü(start)** C. ön sembol(initial)  
D. soyut sembol(abstract)
3. Bir programlama dilinin sözcüksel yapısı, dilin \_\_\_\_ yapısıdır.(The lexical structure of a programming language is the structure of its \_\_\_\_.)  
**A. jetonlarının(tokens)** B. referanslarının(references) C. değişkenlerinin(variables)  
D. açıklamalarının(comments)
4. Bir değişkenin değeri, komutun icra edilmesi dışında değişiyorsa bir \_\_\_\_ oluşmuştur.(When a change in the value of a variable persists beyond the execution of a statement, a(n) \_\_\_\_ has occurred.)  
A. kalıcı etki(persistent effect) **B. yan etki(side effect)** C. bileşik değişim(compound change)  
D. hata(error)
5. Bir ayrıştırıcının en basit hali bir \_\_\_\_.(The simplest form of a parser is a(n) \_\_\_\_.)  
A. yorumlayıcıdır(interpreter) B. çevirmendir(translator) C. derleyicidir(compiler)  
**D. tanıyıcıdır(recognizer)**
6. \_\_\_\_ aşaması esnasında çevirmen/dönüştürücü, giriş programından gelen karakter dizisini alıp bunu jetonlara dönüştürür.(During its \_\_\_\_ phase, a translator collects sequences of characters from the input program and forms them into tokens.)  
A. üretme(generating) B. ayrıştırma(parsing) **C. tarama(scanning)** D. analiz(analyzing)
7. Nesne tabanlı paradigmayı ilk olarak eksiksiz ve tutarlı biçimde dahil eden programlama dili hangisidir?(The first programming language to incorporate the object paradigm in a thorough and consistent way was \_\_\_\_.)  
A. Algol60 B. Fortran-80 C. Simula67 **D. Smalltalk-80**

8. Java programlama dilinde \_\_\_\_ yapıcı metod parametre almaz.(In Java, the \_\_\_\_ constructor takes no parameters.)  
A. aşırı yüklenmiş(overloaded) B. zincirlenmiş(chained) C. baz(base) **D. varsayılan(default)**
9. C++ dilinde üye fonksiyonlar sınıf tanımlamasının dışında \_\_\_\_ işleci kullanılarak uygulanabilir.(Member functions can be implemented outside the declaration of a class in C++ using the \_\_\_\_ operator.)  
**A. kapsam çözümüleme(scope resolution)** B. kapsam değiştirme(scope altering) C. yeniden yönlendirme(redirection) D. bildirim(directive)
10. \_\_\_\_ otomatik olarak BNF tanımlamalarını ayrıştırıcılara dönüştürürler.(\_\_\_\_ automatically translate a BNF description into a parser.)  
A. Sözdizimi üreticileri(Syntax generators) B. Kaydır-azalt ayrıştırıcılar(Shift-reduce parsers) **C. Ayrıştırıcı üreticiler(Parser generators)** D. Tarayıcılar(Scanners)
11. C++ nesnelerin bellekten tahsisinin nasıl kaldırılacağını bildiren \_\_\_\_ içerir.(C++ includes \_\_\_\_, which specify how to deallocate memory for objects.)  
A. tahsis kaldırıcılar(deallocators) B. siliciler(erasers) **C. yıkıcılar(destructors)** D. çöp toplayıcılar(garbage collectors)
12. Java dilinde skaler tipler aynı zamanda \_\_\_\_ tipler olarak da adlandırılır(In Java, scalar data types are also called \_\_\_\_ types.)  
A. niceliksel(quantitative) **B. primitif(primitive)** C. esnek(flexible) D. basit(simple)
13. \_\_\_\_, saklı olan değeri icra etme sırasında değişebilen bir nesnedir.(A \_\_\_\_ is an object whose stored value can change during execution.)  
A. Sabit(constant) B. Metod(method) **C. Değişken(variable)** D. Sınıf(class)
14. Dinamik bellek tahsisi değişkenleri \_\_\_\_ saklar(Dynamic allocation places variables \_\_\_\_.)  
A. yığın alanında(on the stack) B. işaretçi listesinde(in the pointer list) **C. heap alanında(on the heap)** D. sembol tablosunda(in the symbol table)
15. C++ dilinde \_\_\_\_ üyeler türetilen sınıflar tarafından erişilebilir.(In C++, \_\_\_\_ members are accessible by derived classes.)  
**A. korumalı(protected)** B. kısıtlı(restricted) C. özel(private) D. uzatılabilir(extensible)
16. Smalltalk dilinde bir blok aşağıdakilerden hangisi ile çevrelenir?(The Smalltalk block object is enclosed in \_\_\_\_.)  
**A. [ ]** B. ( ) C. { } D. i i
17. Bir işlecin birden fazla işlevi yerine getirmesine \_\_\_\_ denir.(If an operator performs more than one function it is said to be \_\_\_\_.)  
**A. aşırı yükleme(overloaded)** B. dolaylı(implicit) C. değişken(variable) D. gölgeleme(shadowed)

18. Boş alan karakterlerinin işlenmesi genellikle \_\_\_\_ tarafından ele alınır.(The processing of white space is generally handled by a \_\_\_\_.)  
A. ayrıştırıcı üretici(parser generator) B. kestirimci ayrıştırıcı(predictive parser) C. ayrıştırıcı(parser)  
**D. tarayıcı(scanner)**
19. Farklı sınıflardan benzer servisler isteyen mesajlarda aynı adın kullanılmasına \_\_\_\_ denir.(\_\_\_\_ is the use of the same names for messages requesting similar services from different classes.)  
A. ayırım(Distinction) **B. çok biçimlilik(Polymorphism)** C. aşırı gölgeleme(Overshadowing)  
D. genelleme(Generalization)
20. Her bir metod çağrısı \_\_\_\_ olarak adlandırılır.(Each call to a method is referred to as a(n) \_\_\_\_.)  
A. yürütme(invocation) B. icra etme(execution) C. dallanma(branching) **D. etkinleştirme(activation)**
21. \_\_\_\_ daha küçük öbek yapılarına ayrılabilen öbek yapılardır.(\_\_\_\_ are phrase structures that are broken down into further phrase structures.)  
A. Sıkıştırılmış yapılar(Compressed structures) B. Karmaşık gramerler(Complex grammars)  
**C. Nonterminaller(Nonterminals)** D. Terminaller(Terminals)
22. İşaretçiler içlerinde bir \_\_\_\_ saklayan nesnedir.(A pointer is an object whose stored value is a(n) \_\_\_\_.)  
A. yön(direction) B. bileşik değer(compound value) **C. başka nesneye referans(reference to another object)** D. etkinleştirme kaydı(activation record)
23. Çöp alanını otomatik olarak geri alan dil sistemlerine \_\_\_\_ denir.(Language systems that automatically reclaim garbage are said to perform \_\_\_\_.)  
A. belleğin geri istenmesi(memory reclamation) **B. çöp toplama(garbage collection)**  
C. geri dönüşüm(recycling) D. belleğin yeniden tahsisi(memory reallocation)
24. Yazılım bileşenlerinin iç detaylarına erişimin kısıtlanmasına ne ad verilir?(Restricting access to internal details of software components is known as \_\_\_\_.)  
A. soyutlama(abstraction) B. yeniden tanımlama(redefinition) **C. paketleme(encapsulation)**  
D. çok biçimlilik(polymorphism)
25. Çevirmen, \_\_\_\_ aşamasında jetonları işleyerek programın sözdizimsel yapısını belirler.(During the \_\_\_\_ phase, the translator processes the tokens, determining the program's syntactic structure.)  
A. tarama(scanning) B. uygulama(implementation) **C. ayrıştırma(parsing)** D. üretme(generation)

26. (10P) Listede bulunan elemanların karelerini başka bir listeye yazan Prolog programını yazınız.(Write a Prolog program that writes squares of a list to another list.)

```
kareleri ( Liste , Sonuc ).  
kareleri ( [2,5,3,7] , S ).  
S=[4,25,9,49].
```

```
kareleri ( [] , [] ).  
kareleri ( [H|T] , [K|L] ) :- K is H*H, kareleri (T,L).
```

27. (10P) İki listenin elemanlarını toplayan listeTopla fonksiyonunu Haskell dilinde yazın. Fonksiyonun örnek çalışması aşağıda verilmiştir. Toplam listesinin eleman sayısı, eleman sayısı az olan listeninki kadardır.(Write the listeTopla function that calculates the sum of the two lists. Usage of the function is given below. Length of the final list is the same as the length of the list with fewer elements.)

```
*Main> listeTopla [1,2,3] [4,5]  
[5,7]  
*Main> listeTopla [1,2,3] [4,5,6,7]  
[5,7,9]  
*Main> listeTopla [9,8,7,6] [5,4,3,2]  
[14,12,10,8]
```

```
listeTopla [] _ = []  
listeTopla _ [] = []  
listeTopla (x:xs) (y:ys) = (x+y):(listeTopla xs ys)
```

28. Aşağıda verilen C kodu için:

```
1 int x;          7 void f() {          13 h();  
2 int main(){    8     int x=3;      14 }  
3     x=2;        9     h();      15 void h() {  
4     f();        10 }          16     printf("%d\n",x);  
5     g();        11 void g() {      17 }  
6 }              12     int x=4;
```

- (a) (8P) Statik kapsam kullanılırsa kodun çıktısı ne olur?(What is the output when lexical scoping is used?)

```
2 2
```

- (b) (7P) Dinamik kapsam kullanılırsa kodun çıktısı ne olur?(What is the output when dynamic scoping is used?)

3 4

29. (15P) Aşağıdaki grameri düşündüğünüzde S,A ve B nonterminal; a,b,c,t,l ve r ise terminalleri oluşturur. **atlapbpcrta** kelimesi için ayrıştırma ağacını çiziniz.(Consider the grammar below. S,A and B are nonterminals; a,b,c,t,l and r are terminals.Draw the parse tree for the input **atlapbpcrta** )

$S \rightarrow SpA \mid A$

$A \rightarrow BtA \mid B$

$B \rightarrow a \mid b \mid c \mid lSr$

