

	<p style="text-align: center;"><b>T.C.</b> <b>İNÖNÜ ÜNİVERSİTESİ</b> <b>MÜHENDİSLİK FAKÜLTESİ</b> <b>YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ</b> <b>2023/ 2024 EĞİTİM-ÖĞRETİM YILI</b> <b>BAHAR DÖNEMİ FİNAL SINAV KÂĞIDI</b></p>	<p style="text-align: center;"><b>SINAVDA UYULACAK KURALLAR</b></p> <p>-Sınav süresince öğrenci kimlik kartları masanın üzerinde bulundurulacaktır.</p> <p>-Sınav süresince cep telefonlarının açık tutulması yasaktır. Kapalı olsa dahi öğrencinin ulaşabileceği yerlerde bulundurulmayacaktır.</p> <p>-Kulaklık takılması; cep bilgisayar, ses kayıt cihazı, saat fonksiyonu dışında fonksiyonu bulunan saat vb. her türlü iletişim cihazının bulundurulması ve kullanılması yasaktır.</p> <p>-Görevli gözetmenler tarafından sorularla ilgili açıklama yapılmayacaktır.</p> <p>-Sınav düzenini bozan öğrenciler hakkında tutanak tutulacaktır.</p> <p>-Sınav esnasında öğrencilerin konuşması, kalem, silgi vb. malzeme alışverişi yasaktır.</p> <p>-Sınav kurallarını okudum ve kabul ediyorum.</p> <p style="text-align: center;"><b>SINAV SÜRESİ : 50 dk.</b></p>
<p><b>DERSİN KODU ve ADI : 306 - AĞ PROGRAMLAMA</b> <b>TARİH VE SAAT : 01/07/2024 09:00</b></p>		
<b>ÖĞRENCİNİN</b>		
<b>Adı Soyadı</b>		<b>İmzası</b>
<b>Numarası</b>		
		<b>TOPLAM PUAN</b>

### SORULAR

- (5p)** Bir servlet'in *service()* metodu ne zaman çağrılır?  
A. Servlet yüklendiğinde  
B. Servlet ilk başlatıldığında  
C. Servlet sonlandırıldığında  
D. Servlet'e her yeni istek geldiğinde
- (5p)** JSP(Java Server Pages) ile ilgili aşağıdaki ifadelerden hangisi yanlıştır?  
A. JSP dosyaları, HTML, CSS ve JavaScript kodları içerebilir.  
B. **JSP dosyaları, yalnızca sunucu taraflı kodlama için kullanılır.**  
C. JSP dosyaları, Java kodunu HTML içine yerleştirerek dinamik web sayfaları oluşturur.  
D. JSP dosyaları, sunucu tarafında çalıştırılarak dinamik olarak HTML içeriği oluşturur ve kullanıcıya sunar.
- (5p)** ServerSocket'in multi-client(çoklu-istemci) desteği sağlamak için hangi yaklaşım kullanılabilir?  
A. **Her istemci bağlantısında yeni bir Thread oluşturmak**  
B. Her istemci için ayrı bir ServerSocket oluşturmak  
C. TCP/IP bağlantıları için DatagramSocket kullanmak  
D. Her istemci için sunucu tarafında ayrı bir port numarası kullanmak
- (15p)** Aşağıdaki ifadeleri D (Doğru) veya Y (Yanlış) olarak işaretleyiniz.  
☐ **Y** Bir servlet'in *doGet()* ve *doPost()* metodları *init()* metodundan önce çağrılır.  
☐ **D** Servlet/web container, servlet örneklerini yönetir ve yaşam döngülerini kontrol eder.  
☐ **Y** Servlet'ler, web sunucusuna gelen her istekte yeni bir servlet nesnesi/örneği oluşturur.  
☐ **D** *session.invalidate()* metodu, mevcut session'ı sona erdirir ve session içindeki tüm değerleri siler.  
☐ **Y** Cookie'ler, yalnızca sunucular arasında veri taşımak için kullanılır ve istemci tarafında saklanmaz.
- (10p)** Servlet nedir, neden kullanılır? Açıklayınız.  
**Web tabanlı ağ programlamada sunucu tarafında kullanılan bir java programıdır.**  
**Sunucuya gelen Http isteklerini yönetmek için kullanılan bir java sınıfıdır.**  
**Dinamik web sayfaları oluşturmak için kullanılır.**
- (10p)** Servletin yaşam döngüsü adımlarını yazıp her birini kısaca açıklayınız.

- 1.Yükleme (Loading):** Servlet, ilk kez talep edildiğinde veya sunucu başlatıldığında yüklenir ve servlet nesnesi oluşturulur.
- 2.Başlatma (Initialization):** Servlet nesnesi oluşturulduktan sonra *init()* metodu çağrılır. Bu yöntem, servlet'in başlatılmasında gerekli olan tüm başlangıç işlemlerini yapar.
- 3.İstek İşleme (Request Handling):** Her gelen istek için *service()* metodu tetiklenir. *service()* metodu, uygun yöntemi (*doGet()*, *doPost()*, *doPut()*, *doDelete()*) çağırır.
- 4.Sonlandırma (Termination):** Servlet'in yaşam döngüsü sona erdiğinde, *destroy()* metodu çağrılır.

7. (20p) Bir java servlet uygulamasında giriş yapan kullanıcı adını hem session hem de cookie içerisine kaydeden kod parçasını yazınız.

```
String username = request.getParameter("username");

HttpSession session = request.getSession();
session.setAttribute("username", username);

Cookie cookie = new Cookie("username", username);
cookie.setMaxAge(24 * 60 * 60);
response.addCookie(cookie);
```

8. (30p) Aşağıdaki kod parçasında TCP protokolü kullanılarak sadece bir istemciden gelen mesaj alınıp geri gönderilmektedir. Sunucunun aynı anda birden fazla istemci ile haberleşmesini sağlamak için thread kullanılıp çoklu istemci desteği verilmelidir. Buna göre aşağıda verilen kod parçasında gerekli değişiklikleri yapıp kodu tekrar yazınız.

```
public class TCPServer {
    private static ServerSocket serverSocket = null;
    public static void main(String[] args) {
        serverSocket = new ServerSocket(1234);
        Socket clientSocket = serverSocket.accept();
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
        Scanner input = new Scanner(clientSocket.getInputStream());
        while (true) {
            String mesaj = input.nextLine();
            out.println(mesaj);
        }
        serverSocket.close();
    }
}

public class TCPServer {
    private static ServerSocket serverSocket = null;
    public static void main(String[] args) {
        serverSocket = new ServerSocket(1234);
        while (true) {
            Socket clientSocket = serverSocket.accept();
            ClientHandler handler = new ClientHandler(clientSocket);
            handler.start();
        }
        serverSocket.close();
    }
}

public class ClientHandler extends Thread {
    private Socket clientSocket;
    PrintWriter out;
    Scanner input;
    public ClientHandler(Socket clientSocket) {
        this.clientSocket = clientSocket;
    }
    @Override
    public void run() {
        out = new PrintWriter(clientSocket.getOutputStream(), true);
        input = new Scanner(clientSocket.getInputStream());
        while (true) {
            String gelenVeri = input.nextLine();
            System.out.println(clientSocket.toString() + gelenVeri);
            out.println(gelenVeri.toUpperCase());
        }
        clientSocket.close();
    }
}
}
```