

## CSE 222 – HW01 – Q2

*Abstract Data Type*, verilerin soyutlanması demektir. Java'da veriye erişmek için özel metotlar kullanılır. Veri türünü bağımsız kılmak içinde kapsülleme kullanır. Bu Java tarafından yapısal olarak sağlanır. Abstract Data Type'taki asıl amaçlardan birisi programcının/geliştiricinin veri yapısının ve metotlarının nasıl implement edildiğini bilmeden programlama yapabilmesini sağlamaktır. Örneğin stack yapısını ele alalım. Stack oluşturmak için *create\_stack()* adında bir metodumuz olsun. Programcı burda *create\_stack()* metodunun nasıl implement edildiğini bilmek zorunda değildir.

C Programlama dilinde, kompleks verileri gösterebilmek için pointer yapısı kullanılır. Pointer kullanarak verinin tipinden bağımsız olarak fonksiyonlarda verileri aktarabiliriz. Bu bizim tip bağımsız çalışmamıza olanak sağlar.

Buna ek olarak C programlama dilinde implement dosyaları ile header dosyalarını ayırma şansımız vardır. *Header(.h)* dosyası bize fonksiyonların kısa tanımlamalarını(*declaration*), pointer tiplerini ve global erişebilir verileri gördüğümüz yerdir. *Implement (.c)* dosyası ise fonksiyonun tam olarak nasıl yazıldığının görülebileceği dosyasıdır.

C programlama dilinde bir abstract data type oluşturacaksa; bu yapı iki parçadan oluşması gerekir. Header dosyasında kısa fonksiyon tanımlamaları veri tipleri olmalıdır. Implement dosyasında da bu belirtilen fonksiyonlar implement edilmeli ve yapıların nasıl olacağı belirtilmelidir. Örnek vermek gerekirse; *stack* yapısı C dilinde implement edilmesi gerekirse *stack* tipini oluşturmak için *typedef* kullanılır. Verileri bir arada tutmak için *struct* yapısından faydalanılır. *stack*'i oluşturmak için bir *initStack* fonksiyonu yazılır. *Pop* ve *push* işlemleri için de fonksiyonlar yazılır.

C dilinde yapıyı soyutlamak dilin yapısı gereği zordur. Bu kısımda programcının kendisine koşul olarak bu kurallara uyacağını da kabul etmemiz gerekmektedir.