

# 2024-2025

## PROGRAMLAMAYA GİRİŞ

### ÖDEV II

Aşağıda tarif edildiği gibi çalışan bir tavuk çiftliği simülasyonu hazırlamanız istenmektedir.

#### Programın Başlangıcında Yapılacak İşler

Programın başlangıcında aşağıdaki bilgiler kullanıcıdan istenecektir.

- Başlangıç sermayesi
- Çiftlikte kaç tavuk olacağı (en fazla 500)
- Tavuk yeminin kilogram fiyatı
- Yumurtanın satış fiyatı
- Tavuk alım fiyatı
- Simülasyon yapılacak gün sayısı

#### Çiftlik Kurulurken Yapılacak İşler

Çiftlik kurulurken başlangıçta aşağıdaki işler yapılacaktır.

- Kullanıcının girdiği tavuk sayısı kadar tavuk satın alınacak ve bu gider kalan paradan düşülecek.
- 700 kilogram yem alınıp yem ambarına eklenecek ve bu gider kalan paradan düşülecek.

#### Çiftlik Kuralları

Çiftlik aşağıdaki kurallara göre işletilecektir.

- Her tavuk günde 0, 1 veya 2 yumurta yumurtlar. Her tavuk için her gün bu sayı rastgele belirlenir.
- Her tavuk günde 100 ile 120 gram arasında yem tüketir. Her tavuk için her gün bu sayı rastgele belirlenir.
- Çiftlikte yeni gelen bir tavuk ilk 3 gün yumurtlamaz.
- Üst üste 3 gün yumurtlamayan tavuklar kesime gönderilir ve yerine yeni tavuk alınır. Çiftlikte yeni gelen tavukların ilk 3 gün yumurta vermesi beklenmediği için bu günler dikkate alınmaz.
- Yumurtladığı toplam yumurta sayısı 100'e ulaşan tavuklar kesime gönderilir ve yerine yeni tavuk alınır.
- Kesime gönderilen her tavuk için tavuk alım fiyatının yarısı kadar gelir elde edilir. Bu gelir kalan paraya eklenir.
- Kesime gönderilen her tavuğun yerine yeni tavuk alınır. Bu gider kalan paradan düşülsür.
- Kesime gönderilen her tavuğun yerine yenisini alındığı için çiftlikteki tavuk sayısı azalmaz veya artmaz, hep aynı kalır.

## Her Gün Yapılacak İşler

- Bütün tavuklara yem verilecek, tavukların yediği yemler ambardaki yem miktarından düşülecek.
  - Tavuklardan yumurta toplanacak.
  - Son 3 gündür yumurta vermeyen tavuklar kesime gönderilip yerine yeni tavuklar alınacak. Çiftliğe yeni geldiği için yumurtlamayan tavuklar bundan hariç tutulacak.
  - Toplam yumurta sayısı 100'e ulaşan tavuklar kesime gönderilip yerine yeni tavuklar alınacak.
  - Toplanan yumurtalar satılacak ve elde edilen gelir kalan paraya eklenecek.
  - Yem ambarı kontrol edilecek ve 70 kilodan az yem kalmışsa 700 kilo yem satın alınıp ambara eklenecek. Bu gider kalan paradan düşülecek.
  - Günün sonunda kalan para 0'ın altına düşmüşse "İflas ettiniz..." mesajı gösterilerek simülasyon sonlandırılacak.

## Örnek Ekran Görüntüleri

Gün	Yem Tüketimi	Yumurta Kesilen Tavuk	Günlük Gelir	Günlük Gider	Kalan Yem	Kalan Para
1	54974	0	0.00	0.00	645026	58500.00
2	54832	0	0.00	0.00	590194	58500.00
3	55097	0	0.00	0.00	535097	58500.00
4	55039	510	0	2805.00	0.00	480058
5	54953	512	0	2816.00	0.00	425105
6	54951	526	16	3853.00	1920.00	370154
7	55074	487	12	3398.50	1440.00	315080
8	54811	473	5	2901.50	600.00	260269
9	54891	454	13	3277.00	1560.00	205378
10	55111	481	9	3185.50	1080.00	150267
11	54990	446	15	3353.00	1800.00	95277
12	55097	453	10	3091.50	32700.00	740180
13	55116	478	16	3589.00	1920.00	685064
14	54950	470	8	3065.00	960.00	630114
15	54683	464	14	3392.00	1680.00	575431

Gün	Yem Tüketimi	Yumurta	Kesilen	Tavuk	Günlük Gelir	Günlük Gider	Kalan Yem	Kalan Para
1	55058	0	0	0	0.00	0.00	644942	3500.00
2	54909	0	0	0	0.00	0.00	590033	3500.00
3	54814	0	0	0	0.00	0.00	535219	3500.00
4	55015	474	0	0	2607.00	0.00	480204	6107.00
5	55128	515	0	0	2832.50	0.00	425076	8939.50
6	55003	527	22	0	4218.50	2640.00	370073	11838.00
7	54811	477	9	0	3163.50	1080.00	315262	14461.50
8	55043	465	9	0	3097.50	1080.00	260219	17019.00
9	54909	473	10	0	3201.50	1200.00	205310	19620.50
10	54876	456	10	0	3108.00	1200.00	150434	22128.50
11	55051	481	9	0	3185.50	1080.00	95383	24774.00
12	54854	441	10	0	3025.50	32700.00	740529	-4300.50

## Kod Yazımı ile İlgili Kurallar

- Program nesneye dayalı programlama teknikleri ile yazılacaktır.
- Tavuk ve Ciftlik adına iki ayrı sınıf yazılacaktır.
- main fonksiyonunda başlangıç sermayesi, çiftlikte kaç tavuk olacağı, tavuk yeminin kilogram fiyatı, yumurtanın satış fiyatı, tavuk alım fiyatı kullanıcıya sorulacak ve bu bilgiler Ciftlik sınıfının kurucu fonksiyonuna gönderilerek bir Ciftlik sınıfından “ciftlik” nesnesi oluşturulacaktır.
- main fonksiyonunda simülasyon yapılacak gün sayısı kullanıcıya sorulacak ve gün sayısı “ciftlik” nesnesinin “simulasyonYap” fonksiyonuna parametre olarak gönderilerek simülasyon yaptırılacaktır.
- Her tavuk nesnesi kaç gündür yumurtlamadığını, çiftliğe kaç gün önce geldiğini ve çiftliğinden beri toplam kaç yumurta verdiği private üyelerde kendi saklayacaktır.
- Her tavuk nesnesinin “yumurtla” fonksiyonu çağrılarak günlük olarak tavuktan yumurta alınacaktır.
- Her tavuk nesnesinin “yemYe” fonksiyonu çağrılarak günlük olarak tavuğa yem verilecektir.
- Çiftlikteki tavuklar Ciftlik sınıfının içinde dizi şeklinde saklanacaktır.
- main fonksiyonunda oluşturulan ciftlik nesnesi kalan para, tavuk sayısı, kalan yem, yem kilogram fiyatı, yumurta fiyatı ve tavuk alım fiyatı bilgilerini private üyelerde kendi saklayacaktır.

Örnek main fonksiyonu aşağıda verilmiştir.

```
int main(int argc, char* argv[]) {
    // TODO: Başlangıç işlemleri yapılacak
    // TODO: Değişkenler tanımlanacak
    // TODO: Kullanıcıdan bilgiler istenecek

    Ciftlik ciftlik(sermaye, tavukSayisi, yemKgFiyati, yumurtaFiyati, tavukAlimFiyati);
    ciftlik.simulasyonYap(gunSayisi);

    return 0;
}
```

## Teslim Kuralları

- Tüm program .cpp uzantılı tek bir dosyanın içine yazılacaktır.
- Ödevde yapılan işlemlerin özet bir şekilde anlatıldığı bir rapor dosyası Word veya PDF formatında hazırlanacaktır.
- Kaynak kodu dosyası ile rapor dosyası (toplam 2 dosya) zip formatında sıkıştırılarak SABİS'e yüklenecektir.
- SABİS'e yüklenmeyen ödevler değerlendirilmeyecektir.
- Zamanında teslim edilmeyen ödevler değerlendirilmeyecektir.
- Ödevin son teslim zamanı SABİS'te belirtildiği gibidir.

# UYULMASI GEREKEN KOD DÜZENİ

Her C++ dosyasının başında aşağıdaki yorum bloğu bulunacaktır. Yorum bulunmayan her C++ dosyası için 10 puan kırılacaktır. (PDF üzerinden kopyalayıp yapıştırmanız problem çıkarabilir)

```
*****  
**  
** SAKARYA ÜNİVERSİTESİ  
** BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ  
** BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
** PROGRAMLAMAYA GİRİŞİ DERSİ  
**  
** ÖDEV NUMARASI.....:  
** ÖĞRENCİ ADI.....:  
** ÖĞRENCİ NUMARASI...:  
** DERS GRUBU.....:  
*****/
```

Değişken isimleri anlamlı olmalıdır. Örneğin tek sayı adedini tutacak bir değişken için

YANLIŞ
int a;

yerine

DOĞRU
int tekAdedi;

yazılmalıdır.

Her bir küme parantezinin altındaki kodları yazmadan önce **tab** tuşuna basılarak boşluk bırakılmalıdır.

YANLIŞ
if(true) { int a =6; cout<<"Merhaba"; if(false) cout<<"Nasilsin"; }

DOĞRU
if(true) { int a =6; cout<<"Merhaba"; if(false) cout<<"Nasilsin"; }

Her döngü ve koşul işlemlerinden önce yapılan işlem hakkında yorum yazılmalıdır

YANLIŞ
int t = 0; for(int i=0;i<100;i++) t+=i;

DOĞRU
int toplam = 0; //1 den 100 e kadar olan sayılar toplanıyor for(int i=0;i<100;i++) toplam+=i;

# ÖDEV TESLİM KURALLARI

## DERLEYİCİ VE TESLİM TARİHİ

Derleme için **Visual Studio 2019** veya daha güncel C++ derleyicisi kullanılacaktır.

Zamanında teslim edilmeyen ödevler değerlendirilmeyecektir.

Teslim saati ve günü sisteme belirtilmiştir

# 2024-2025

## INTRODUCTION TO PROGRAMMING

### ASSIGNMENT II

You are required to prepare a chicken farm simulation that operates as described below.

#### Initial Tasks for the Program

At the start of the program, the following information will be requested from the user:

- Initial capital
- Number of chickens in the farm (maximum 500)
- Price per kilogram of chicken feed
- Sale price of an egg
- Purchase price of a chicken
- Number of days for the simulation

#### Tasks During Farm Setup

During the initial setup of the farm, the following actions will be performed:

- The number of chickens specified by the user will be purchased, and this expense will be deducted from the remaining balance.
- 700 kilograms of feed will be purchased and added to the feed storage, and this expense will also be deducted from the remaining balance.

#### Farm Rules

The farm will operate according to the following rules:

- Each chicken lays 0, 1, or 2 eggs per day. This number is randomly determined for each chicken daily.
- Each chicken consumes between 100 and 120 grams of feed per day. This amount is randomly determined for each chicken daily.
- A newly purchased chicken will not lay eggs for the first 3 days.
- Chickens that do not lay eggs for 3 consecutive days are sent for slaughter and replaced with new chickens. The 3-day non-laying period for newly purchased chickens is not considered.
- Chickens that reach a total of 100 eggs laid are sent for slaughter and replaced with new chickens.
- For each chicken sent for slaughter, revenue equivalent to half the chicken purchase price is earned, which is added to the remaining balance.
- A new chicken is purchased for every slaughtered chicken, so the number of chickens on the farm neither increases nor decreases and remains constant.

## Daily Tasks

- All chickens are fed, and the consumed feed is subtracted from storage.
  - Eggs are collected from the chickens.
  - Chickens that have not laid eggs for the last 3 days are sent for slaughter, and new chickens are purchased in their place. Chickens that do not lay eggs due to the 3-day adjustment period are excluded from this rule.
  - Chickens with a total egg count of 100 are sent for slaughter, and new chickens are purchased in their place.
  - The collected eggs are sold, and the revenue is added to the remaining balance.
  - The feed storage is checked, and if less than 70 kilograms of feed remains, 700 kilograms of feed are purchased and added to the storage. This expense is deducted from the remaining balance.
  - If the remaining balance falls below 0 at the end of the day, the simulation ends with the message "You have gone bankrupt...".

## Sample Screen Outputs

Day	Feed Consumption	Eggs	Slaughtered	Daily Revenue	Daily Expense	Remaining Feed	Feed Balance
1	54974	0	0	0.00	0.00	645026	58500.00
2	54832	0	0	0.00	0.00	590194	58500.00
3	55097	0	0	0.00	0.00	535097	58500.00
4	55039	510	0	2805.00	0.00	480058	61305.00
5	54953	512	0	2816.00	0.00	425105	64121.00
6	54951	526	16	3853.00	1920.00	370154	67014.00
7	55074	487	12	3398.50	1440.00	315080	69692.50
8	54811	473	5	2901.50	600.00	260269	72294.00
9	54891	454	13	3277.00	1560.00	205378	74791.00
10	55111	481	9	3185.50	1080.00	150267	77436.50
11	54990	446	15	3353.00	1800.00	95277	79889.50
12	55097	453	10	3091.50	32700.00	740180	50881.00
13	55116	478	16	3589.00	1920.00	685064	53510.00
14	54950	470	8	3065.00	960.00	630114	56095.00
15	54683	464	14	3392.00	1680.00	575431	58647.00

Day	Feed Consumption	Eggs	Slaughtered	Daily Revenue	Daily Expense	Remaining Feed	Balance
1	55058	0	0	0.00	0.00	644942	3500.00
2	54909	0	0	0.00	0.00	590033	3500.00
3	54814	0	0	0.00	0.00	535219	3500.00
4	55015	474	0	2607.00	0.00	480204	6107.00
5	55128	515	0	2832.50	0.00	425076	8939.50
6	55003	527	22	4218.50	2640.00	370073	11838.00
7	54811	477	9	3163.50	1080.00	315262	14461.50
8	55043	465	9	3097.50	1080.00	260219	17019.00
9	54909	473	10	3201.50	1200.00	205310	19620.50
10	54876	456	10	3108.00	1200.00	150434	22128.50
11	55051	481	9	3185.50	1080.00	95383	24774.00
12	54854	441	10	3025.50	32700.00	740529	-4300.50

## Code Writing Rules

- The program must be written using object-oriented programming techniques.
- Two separate classes will be written for Chicken and Farm.
- In the main function, the following information will be requested from the user: initial capital, the number of chickens in the farm, the price per kilogram of chicken feed, the sale price of eggs, the purchase price of chickens, and the number of days for the simulation. This information will be passed to the constructor of the Farm class to create a farm object.
- In the main function, the number of days for the simulation will be asked from the user, and this value will be passed as a parameter to the simulate function of the farm object to run the simulation.
- Each chicken object will keep track of how many days it has not laid eggs, how many days it has been on the farm, and how many eggs it has laid since joining the farm as private member variables.
- Each chicken object will have a layEgg function that will be called to collect eggs daily.
- Each chicken object will have a consumeFeed function that will be called to feed the chicken daily.
- The chickens on the farm will be stored in an array within the Farm class.
- The farm object created in the main function will keep track of the remaining money, the number of chickens, the remaining feed, the price per kilogram of feed, the price of eggs, and the purchase price of chickens as private member variables.

Example main function is as follows:

```
int main(int argc, char* argv[]) {  
    // TODO: Initialize the program  
    // TODO: Declare variables  
    // TODO: Request user input  
  
    Farm farm(initialCapital, numberOfChickens, feedPricePerKg, eggPrice, chickenPurchasePrice);  
    farm.simulate(numberOfDays);  
  
    return 0;  
}
```

## Submission Rules

- The entire program must be written in a single .cpp file.
- A report file summarizing the tasks performed in the assignment should be prepared in Word or PDF format.
- Both the source code file and the report file (total of 2 files) must be compressed in a ZIP format and uploaded to SABİS.
- Assignments that are not uploaded to SABİS will not be evaluated.
- Assignments submitted after the deadline will not be evaluated.
- The assignment's deadline is specified in SABİS.

# CODE FORMATTING RULES TO FOLLOW

Each C++ file must begin with the following comment block. Any C++ file without this comment will result in a 10-point deduction. (Copy-pasting from the PDF may cause issues.)

```
*****  
** SAKARYA UNIVERSITY  
** FACULTY OF COMPUTER AND INFORMATION SCIENCES  
** DEPARTMENT OF COMPUTER ENGINEERING  
** INTRODUCTION TO PROGRAMMING COURSE  
**  
** ASSIGNMENT NUMBER....:  
** STUDENT NAME.....:  
** STUDENT NUMBER.....:  
** COURSE GROUP.....:  
*****/
```

Variable names should be meaningful. For example, for a variable that holds the count of odd numbers, use a name like oddCount instead of a generic name like a.

WRONG	CORRECT
int a;	int oddCount;

Before writing the code under each block of braces, a tab should be pressed to insert an indentation.

WRONG	CORRECT
if (true) { int a =6; cout << "Hello"; if (false) cout<<"How are you?"; }	if (true) { int a =6; cout << "Hello";  if (false) cout<<"How are you?"; }

A comment must be written before each loop and conditional operation explaining the task being performed.

WRONG	CORRECT
int t = 0; for(int i=0;i<100;i++) t+=i;	int total = 0; // Adding numbers from 1 to 100 for(int i=0;i<100;i++) total+=i;

# ASSIGNMENT SUBMISSION RULES

## Compiler and Submission Date

- The compiler for this assignment must be Visual Studio 2019 or a newer C++ compiler.
- Assignments submitted after the deadline will not be evaluated.
- The submission time and date are specified in the system.

## Code Sharing is Prohibited

Assignments are individual, and all assignments will be compared. Assignments that are too similar will be considered plagiarism.

In the case of plagiarism, the assignment will not be evaluated, and necessary investigations and penalties will be initiated.

## Important Note

The three upload attempts given to you in the system are not for three different files. Only the most recently uploaded file will be kept in the system. Please upload your assignment in one go after completing it, not in parts.

# KOD PAYLAŞMAK YASAKTIR

Ödevler bireyseldir ve verilen bütün ödevler karşılaşılacaktır. Birbirine çok benzeyen ödevler **kopya** muamelesi görecektir. Öğretim üyesi kopya durumunda ödevi değerlendirmez veya gerekli soruşturma ve ceza işlemlerini başlatabilir.

**Önemli NOT:** Sistemde size verilen üç defa yükleme hakkı üç farklı dosya için değildir. Sadece son yüklediğiniz dosya sistemde tutulmaktadır. Ödevinizi tamamen bitirdikten sonra parça parça değil tek seferde yükleyiniz.