

ÖDEV 3:

(CEVAPLAR)

1. Aşağıdaki programın çıktısını yazınız.

<pre>#include <iostream> #include<string> using namespace std; class Student { public: Student():status(1),name(""){ age = 19; cout << "Constructor1" << endl; } Student(int i,string n) :status(2){ Name=n; age = i; cout << "Constructor2" << endl; } ~Student(){cout << "Destructor" << endl;} void show() const; private: const int status; int age; string name; };</pre>	<pre>void Student::show() const { cout << "Status: " << status; cout << "Name:" << name; cout << ", age:" << age << endl; } int main(){ Student st1(20, "Ali"); { Student st2; st2.show(); } st1.show(); return 0; }</pre>
---	---

Cevap:

```
Constructor2
Constructor1
Status: 1Name :, age:19
Destructor
Status: 2Name :Ali, age:20
Destructor
```

2. Aşağıdaki programı ele alalım.

```
#include <iostream>
using namespace std;

class Battery{
public:
    Battery(double =0.0,double =0.2);
    double getPower(){return power;}
    void discharge(double duration);
    void charge(){power=maxPower;}
    ~Battery(){cout<<"Battery Destructor"<<endl;}
private:
    double maxPower;
    double dischargeRate;
    double power;
};
```

```

Battery::Battery(double maxpower, double dischargerate) {
    maxPower=maxpower;
    dischargeRate=dischargerate;
    power=maxPower;
    cout<<"Battery constructor."<<endl;
}
void Battery::discharge(double duration){
    power=power-(duration*dischargeRate);
    if(power<0)
        power=0;
}

int main(){
    Battery batt1(2000,0.20);
    const Battery batt2(1500,0.10);

    batt1.discharge(5000);
    cout<<"Current power of batt1 is "<<batt1.getPower()<<endl;
    batt1.charge();
    cout<<"Current power of batt1 is "<<batt1.getPower()<<endl;
    cout<<"Current power of batt2 is "
        <<batt2.getPower()<<endl;
    return 0;
}

```

- Hangi kod satırlarında derleme hatası vardır? Hataların nedenini açıklayınız.
- Hataı düzeltiniz. Main fonksiyonda bir değişiklik yapmayınız. Sadece Battery sınıfı üzerinde değişiklik yapabilirsiniz.
- Hata düzeltildikten sonra, program çıktısını yazınız.

Cevap:

- (a) Hata 1: Hatalı sabit değişken değer ataması. Atama işlemi “constructor base/member initializer list” içinde yapılmalıdır.

Hata 2: Sabit bir nesnenin ancak sabit fonksiyonları çağrılabilir. batt2, sabit bir nesne olduğundan çağrı yapılan getPower() adlı üye fonksiyonun da sabit fonksiyon olması gerekir.

- (b) Hata 1 ‘in düzeltilmesi için kodlama:

```

Battery::Battery(double maxpower, double dischargerate)
    :maxPower(maxpower), dischargeRate(dischargerate) {

    power=maxPower;
    cout<<"Battery constructor."<<endl;
}

```

Hata 2’nin düzeltilmesi için kodlama:

```

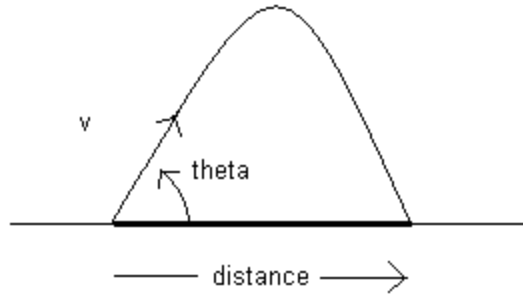
double getPower() const {return power;}

```

(c)

```
Battery constructor.  
Battery constructor.  
Current power of batt1 is 1000  
Current power of batt1 is 2000  
Current power of batt2 is 1500  
Battery Destructor.  
Battery Destructor.
```

3. Savunma sanayisinde kullanılmak üzere bir yazılım modülü istenmektedir. Bu modül, fırlatılacak roketlerin hızı ve atış açısı verildiğinde menzilini hesaplamaktadır. Birimler, hız için metre/saniye, açı için derece, menzil için metre olarak kullanılmaktadır. Yüzeyin yatay ve düz olduğu, yerçekimi ivmesinin de 9.8 m/s^2 olduğu ve hava direncinin olmadığı kabul edilecektir.



Roket adında bir sınıf kodlayınız. Sınıf, hız, aci ve yerçekimiIvmesi adlarında private üye değişkenler içerecektir. Ayrıca,

- setHiz, getHiz, setAci, getAci fonksiyonlarını uygun parametreler ile içerecektir.
- print fonksiyonu içerecek, bu fonksiyon aşağıdaki çıktıyı ekranda gösterecektir. xxx yerine menzil değeri yazılacaktır.

```
Roket, xxx metre uzağa gidecektir
```

- menzilHesapla fonksiyonu yazınız. Bu fonksiyon, menzili hesaplayıp döndürecek. Menzilin hesaplanması için kullanılacak formül internetten bulunabilir.

Yukarıdaki sınıf kodları, roket.h ve roket.cpp adlı kaynak dosyalarda bulunacaktır.

Bir uygulama programı yazınız. Bu programa roketTest.cpp adını veriniz. Bu program, kullanıcıdan, hız ve açı girmesini isteyecektir. Daha sonra da, hesaplanan menzil ekrana yazdırılacaktır. Aşağıda örnek ekran çıktısı vardır.

Roket hızını giriniz (m/s) : 25.7 ↵
Roket fırlatma açısını giriniz (derece) : 64.5 ↵
Roket, 52.3773 metre uzağa gidecektir

Cevap:

roket.h

```
#define PI 3.14
```

```
class Roket
```

```
{
```

```
public:
```

```
    Roket(float = 0, float = 0, float = 9.8) ;
```

```
    float menzilHesapla();
```

```
    void setHiz(float h);
```

```
    float getHiz();
```

```
    void setAci(float a);
```

```
    float getAci();
```

```
    void print();
```

```
private:
```

```
    float hiz;
```

```
    float aci;
```

```
    const float yercekimiIvmesi;
```

```
};
```

roket.cpp

```
#include "roket.h"
```

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
Roket::Roket(float a, float h, float yercekimi):aci(a),hiz(h),yercekimiIvmesi(yercekimi)
{}
```

```
float Roket::menzilHesapla()
```

```
{
```

```
    return (2*hiz*hiz*sin((float)(aci*PI/180))*cos((float)(aci*PI/180))/yercekimiIvmesi);
```

```
}
```

```
void Roket::setHiz(float h){
```

```
    hiz = h;
```

```
}
```

```
float Roket::getHiz(){
```

```
    return hiz;
```

```
}
```

```
void Roket::setAci(float a){
```

```
    aci = a;
```

```
}
```

```
float Roket::getAci(){
```

```
    return aci;
```

```
}
```

```
void Roket::print(){
```

```
    cout << "Roket," << menzilHesapla() << " metre uzaga gidecektir" << endl;
```

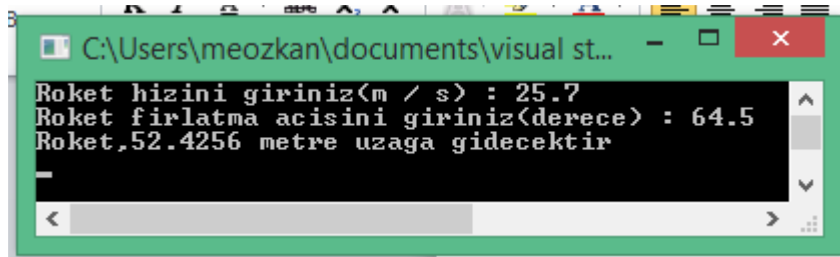
```
}
```

roketTest.cpp

```
#include "roket.h"
#include <iostream>
using namespace std;

int main(){
    Roket roket;
    float hiz, aci;
    cout << "Roket hizini giriniz(m / s) : ";
    cin >> hiz;
    roket.setHiz(hiz);
    cout << "Roket fırlatma acisini giriniz(derece) : ";
    cin >> aci;
    roket.setAci(aci);

    roket.print();
    return 0;
}
```



4. Invoice (fatura) adında bir sınıf yaratınız. (Kaynak dosyalar, invoice.h ve invoice.cpp olacak.)

- En az 4 adet üye değişkene sahip olacaktır. itemNumber (ürün numarası), itemDescription (ürün tanımı), quantity (satın alınan ürün miktarı) ve price (ürün birim fiyatı)
- İki constructor fonksiyonu olacaktır: Biri parametre içermeyecek ve üye değişkenlere varsayılan değerler atayacaktır. Diğer constructor, her bir üye değişken için bir değer parametre olarak alacak ve üye değişkenlere atıyacaktır.
- Üye değişkenler için erişim (accessor) ve değişim (mutator) fonksiyonlar olacaktır. (setPrice, getPrice, v.b.)
- getFee adında bir üye fonksiyon olacaktır. Bu ödenecek ücreti (fee) hesaplayacak (ürün miktarı ile fiyatının çarpımı olarak) ve hesaplanan değeri döndürecektir.
- Bu sınıfı kullanan bir program yazınız. (Kaynak dosya adı: invoiceTest.cpp olsun.) Bu program ürün miktarını soracak, daha sonra ürün bilgilerinin girilmesini isteyecek ve fatura dökümü ve ödenecek bedeli yazdıracaktır. Aşağıdaki örneğe benzer bir çıktı üretilcektir.

```
Enter the number of products: 3 ↵
Enter the product number: 1234168 ↵
Enter the product name: WII ↵
```

```
Enter the product price: 7000 ↵
Enter the product number: 1234268↵
Enter the product name: Digital Camera↵
Enter the product price: 8000↵
Enter the product number: 1234368↵
Enter the product name: iPod Nano↵
Enter the product price 6000↵
=====
Enter the quantity of WII: 2↵
Enter the quantity of Digital Camera: 1↵
Enter the quantity of iPod Nano: 1↵
-----
1234168  WII                7000    2
1234268  Digital Camera    8000    1
1234368  iPod Nano         6000    1
The total fee is 28000.
```

Cevap:

invoice.h

```
class Invoice{
public:
    Invoice();
    Invoice(int iNumber, string iDescription, int q, float p);
    void setItemNumber(int iNumber);
    int getItemNumber();
    void setitemDescription(string iDescription);
    string getItemDescription();
    void setQuantity(int q);
    int getQuantity();
    void setPrice(float p);
    float getPrice();
    float getFee();
private:
    int itemNumber;
    string itemDescription;
    int quantity;
    float price;
};
```

```
invoice.cpp
#include "Invoice.h"
#include <iostream>
#include<string>
using namespace std;

Invoice::Invoice():itemNumber(0), itemDescription(""), quantity(0),
price(0)
{}

Invoice::Invoice(int iNumber, string iDescription, int q, float p)
    :itemNumber(iNumber), itemDescription(iDescription)
    , quantity(q), price(p)
{}

void Invoice::setItemNumber(int iNumber)
{
    itemNumber = iNumber;
}

int Invoice::getItemNumber(){
    return itemNumber;
}

void Invoice::setItemDescription(string iDescription){
    itemDescription = iDescription;
}

string Invoice::getItemDescription(){
    return itemDescription;
}

void Invoice::setQuantity(int q){
    quantity = q;
}

int Invoice::getQuantity(){
    return quantity;
}

void Invoice::setPrice(float p){
    price = p;
}

float Invoice::getPrice(){
    return price;
}

float Invoice::getFee(){
    return (quantity*price);
}
```

invoiceTest.cpp

```

#include "Invoice.h"
#include <iostream>
#include<string>
using namespace std;

int main(){
    Invoice *invoice;
    int productNumber;
    int itemNumber,quantity;
    float price;
    string itemDescription;

    cout << "Enter the number of products : ";
    cin >> productNumber;
    invoice = new Invoice[productNumber];

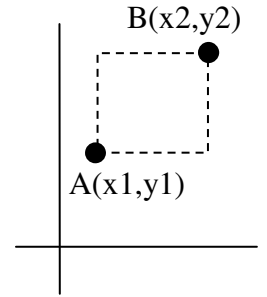
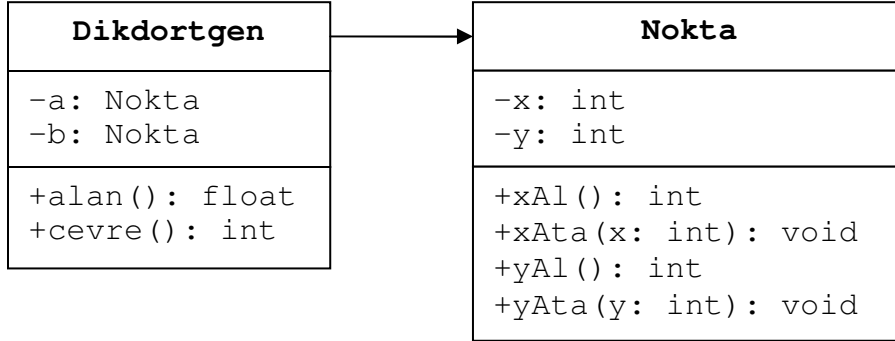
    for (int i = 0; i < productNumber; i++){
        cout << "Enter the product number : ";
        cin >> itemNumber;
        invoice[i].setItemNumber(itemNumber);

        cout << "Enter the product name : ";
        cin >> itemDescription;
        invoice[i].setItemDescription(itemDescription);

        cout << "Enter the product price : ";
        cin >> price;
        invoice[i].setPrice(price);
    }
    cout<<"== == == == == == == == == == == == == == == == == ==<< endl;
    for (int i = 0; i < productNumber; i++){
        cout<<"Enter the quantity of "
            << invoice[i].getItemDescription()<<" : ";
        cin >> quantity;
        invoice[i].setQuantity(quantity);
    }
    cout << "----- -" << endl;
    float totalFee = 0;
    for (int i = 0; i < productNumber; i++){
        cout << invoice[i].getItemNumber() << " "
            << invoice[i].getItemDescription() << " "
            << invoice[i].getPrice() << " "
            << invoice[i].getQuantity() << endl;
        totalFee += invoice[i].getFee();
    }
    cout << "The total fee is " << totalFee<<endl;
    delete [] invoice;
    return 0;
}

```


5. Aşağıda şekilde gösterilen UML diyagrama göre, Nokta (Nokta.h, Nokta.cpp) ve Dikdortgen (Dikdortgen.h, Dikdortgen.cpp) sınıflarını, aşağıda verilen uygulama programı ve çıktıya uygun olarak kodlayınız. Programınızı derleyip, çalıştırıp test etmeyi ihmal etmeyin.



```

#include <iostream>
#include "Nokta.h"
#include "Dikdortgen.h"
using namespace std;

int main() {

    int x, y;
    cout << "A noktası (x,y): ";
    cin >> x >> y;
    Nokta A(x, y);
    cout << "B noktası (x,y): ";
    cin >> x >> y;
    Nokta B(x, y);
    Dikdortgen d(A, B);
    cout << "Cevre: " << d.cevre() << endl;
    cout << "Alan : " << d.alan() << endl;
    return 0;
}
  
```

Örnek çıktı

```

A noktası (x,y):4 5 ↵
B noktası (x,y):10 15 ↵
Cevre: 32
Alan : 60
  
```

Cevap:

Nokta.h

```
class Nokta{
public:
    Nokta(int _x=0, int _y=0) : x(_x), y(_y) {}
    int xAl();
    void xAta(int _x);
    int yAl();
    void yAta(int _y);
private:
    int x;
    int y;
};
```

Nokta.cpp

```
#include "nokta.h"

int Nokta::xAl()
{
    return x;
}

void Nokta::xAta(int _x)
{
    x=_x;
}

int Nokta::yAl()
{
    return y;
}

void Nokta::yAta(int _y)
{
    y=_y;
}
```

Dikdortgen.h

```
#include "nokta.h"

class Dikdortgen{
public:
    Dikdortgen(Nokta &_a, Nokta &_b) : a(_a), b(_b) {}
    float alan();
    int cevre();
private:
    Nokta a;
    Nokta b;
};
```

Dikdortgen.cpp

```
#include "Dikdortgen.h"

float Dikdortgen::alan()
{
    int kenar1=abs(a.xA1()-b.xA1());
    int kenar2=abs(a.yA1()-b.yA1());
    return (kenar1*kenar2);
}

int Dikdortgen::cevre()
{
    int kenar1=abs(a.xA1()-b.xA1());
    int kenar2=abs(a.yA1()-b.yA1());
    return 2*(kenar1+kenar2);
}
```