

152113017: NESNE TABANLI PROGRAMLAMA I
ARA SINAV (23.11.2016)

Kitap ve notlar kapalı

Toplam: 100 puan, 4 Soru, Süre: 90 dakika

S.1	S.2	S.3	S.4	Toplam

Adı-Soyadı: CEVAPLAR

Numara: CEVAPLAR

S.1 (30p.) Aşağıdaki programların çıktılarını yazınız.

```
#include <iostream>
using namespace std;
int f();
int i;
int main() {
    double i = 3.14;
    ::i = f();
    cout << i << " " << ::i << " ";
    return 0;
}
int f()
{
    char i = '*';
    ::i = 6;
    cout << i << " " << ::i << " ";
    return ::i;
}
```

Cevap:

*** 6 3.14 6**

```
#include <iostream>
using namespace std;
void f(int a, int b) { cout << a + b << " ";
}
void f(int a, int b, int c) { cout << a + b +
c << " "; }
void f(int a, double b, int c = 0) { cout <<
(a + c)*b << " "; }
int main() {
    int x = 3, y = 7;
    double z = 3.0;
    f(x, y);
    f(x, y, 6);
    f(x, z);
    return 0;
}
```

Cevap:

10 16 9

```
#include <iostream>
using namespace std;
namespace alan1 {
    int x = 9;
    float y = 3.1;
}
namespace alan2 {
    double x = 29;
    char y = 'a';
}
int main() {
    using namespace alan1;
    cout << x << " ";
    cout << alan2::x << " ";
    cout << alan2::y << " ";
    cout << y << " ";
    return 0;
}
```

Cevap:

9 29 a 3.1

```
#include <iostream>
using namespace std;
int a = 6, b = 10, c = 12;
void fun(int a, int &b, int c = 8) {
    b *= 2;
    a += 3;
    c += 5;
    cout << a << b << c << endl;
}
void fun(int &a) {
    a *= 2;
    c += ::a;
    cout << a << b << c << endl;
}
int main() {
    fun(b);
    fun(b, a);
    fun(b, c, a);
    return 0;
}
```

Cevap:

202018

231213

233617

```
#include <iostream>
using namespace std;
int main() {
    int *p1 = new int[5];
    int *p2;
    for (int i = 0; i < 5; i++) {
        p1[i] = i + 5;
    }
    p2 = p1;
    (*p1)++;
    p2++;
    cout << p1[0] << " " << (*p2);
    return 0;
}
```

Cevap:

6 6

S.2 (20p.) Aşağıda verilen programı nesne tabanlı yaklaşım ile tekrar yazınız. (Önemli: Nesne tabanlı yaklaşımdaki temel kurallara (encapsulation, data hiding, v.b.) mutlaka riayet ediniz. Çalışır bir program yazmanız puan kazandırmaz. Nesne tabanlı yaklaşımdaki kurallara uyulmazsa not alamazsınız.)

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

string urunAdi="tanimsiz";
float urunBirimFiyati = 0;
int urunMiktari = 0;

void urunGoster(string adi, int miktar
                , float birimFiyat){
    cout << adi<<" , "<<miktar<<" birim, "
         <<birimFiyat<<" TL/birim"<<endl;
}
float toplamFiyati(int miktar,float birimFiyat)
{
    return (miktar*birimFiyat);
}
int main() {
    urunAdi = "kitap";
    urunBirimFiyati = 25;
    urunMiktari = 3;
    urunGoster(urunAdi
                ,urunMiktari, urunBirimFiyati);
    cout << "Toplam Fiyati: "
    << toplamFiyati(urunMiktari, urunBirimFiyati)
         <<" TL."<<endl;
    return 0;
}
```

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

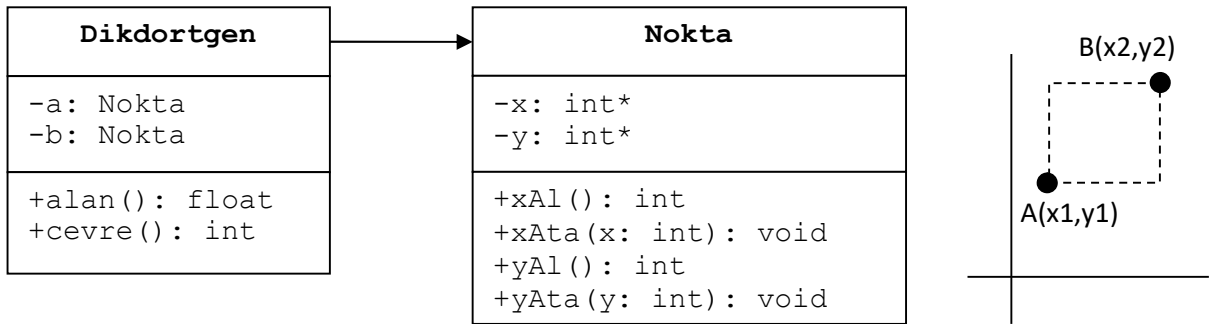
class Urun {
public:
    Urun(string _adi = "tanimsiz"
        ,float _birimFiyat = 0, int _miktar = 0){
        adi = _adi;
        birimFiyat = _birimFiyat;
        miktar = _miktar;
    }

    void goster()
    {
        cout << adi << " , "
        << miktar << " birim, "
        << birimFiyat
        << " TL/birim" << endl;
    }

    float toplamFiyati()
    {
        return (miktar*birimFiyat);
    }
private:
    string adi;
    float birimFiyat;
    int miktar;
};

int main() {
    Urun urun("kitap", 25, 3);
    urun.goster();
    cout << "Toplam Fiyati: "
        << urun.toplamFiyati()
        <<" TL."<<endl;
    return 0;
}
```

S.3 (25p.) Aşağıda şekilde gösterilen UML diyagrama göre, Nokta (Nokta.h, Nokta.cpp) sınıfının kodları verilmektedir.



Nokta.h

```
class Nokta {
public:
    Nokta(int _x = 0, int _y = 0) {
        x = new int(_x);
        y = new int(_y);
        cout<< "Nokta Constructor"<<endl;
    }
    ~Nokta(){
        cout<<"Nokta Destructor" << endl;
    }
    int xA1();
```

Nokta.cpp

```
#include "nokta.h"
int Nokta::xA1()
{
    return (*x);
}
void Nokta::xAta(int _x)
{
    (*x) = _x;
}
int Nokta::yA1()
{
```

<pre> void xAta(int _x); int yAl(); void yAta(int _y); private: int *x; int *y; }; </pre>	<pre> return (*y); } void Nokta::yAta(int _y) { (*y) = _y; } </pre>
---	---

(i) (a) Aşağıdaki main fonksiyonunun çıktısı nedir? (b) Çıktıdaki sorun nedir? (c) Bu sorunu gidermek için gereken üye fonksiyonu yazınız.

```

int main() {
    Nokta nokta1(1,2);
    Nokta nokta2 = nokta1;
    nokta2.xAta(7);
    cout<<nokta1.xAl()<<" "<<nokta1.yAl()<<" " << nokta2.xAl()<<" " <<nokta2.yAl()<< endl;
    return 0;
}

```

Cevap:

- (a) Nokta Constructor
7 2 7 2
Nokta Destructor
Nokta Destructor
- (b) nokta2 yaratılırken, varsayılan kopya yapıcı fonksiyon çağrıldı. x ile y pointerlerinin değerleri yerine adresleri kopyalandı. Bu nedenle, nokta1 ve nokta2 nesnelerinin x ve y pointerleri aynı değişkeni işaret etmektedir. Kopya yapıcı fonksiyon yazmak gerekmektedir.
- (c) Nokta::Nokta(const Nokta &n) {
 x=new int(n.x); y=new int(n.y);
}

(ii) (a) Dikdörtgen sınıfını yazınız. Dikdörtgen sınıfının yapıcı fonksiyonu iki noktayı parametre olarak alacaktır ve “Dikdortgen constructor” yazdıracaktır. Yıkıcı fonksiyonu ise, sadece “Dikdortgen destructor” yazdıracaktır. (b) (i) şıkında, hatanın giderildiği Kabul edilirse ve Dikdortgen sınıfının (a) şıkında istendiği gibi doğru yazıldığı Kabul edilirse, aşağıdaki programın çıktısı ne olur?

```

int main() {
    Nokta nokta1(1,2), nokta2(3, 4);
    Dikdortgen dikdortgen(nokta1, nokta2);
    return 0;
}
(a)
class Dikdortgen {
public:
    Dikdortgen(Nokta &_a, Nokta &_b)
        :a(_a), b(_b) {
        cout << "Dikdortgen Constructor"
            << endl;
    }
    ~Dikdortgen() {
        cout << "Dikdortgen Destructor"
            << endl;
    }
    float alan();
    int cevre();
private:
    Nokta a;
    Nokta b;
};
float Dikdortgen::alan()
{
    int kenar1 = abs(a.xAl() - b.xAl());
    int kenar2 = abs(a.yAl() - b.yAl());
    return (kenar1*kenar2);
}
int Dikdortgen::cevre()
{
    int kenar1 = abs(a.xAl() - b.xAl());
    int kenar2 = abs(a.yAl() - b.yAl());
    return 2 * (kenar1 + kenar2);
}

```

(b)

Nokta Constructor
Nokta Constructor
Dikdortgen Constructor
Dikdortgen Destructor
Nokta Destructor
Nokta Destructor
Nokta Destructor
Nokta Destructor

S.4 (25p.) (a) Aşağıda UML diyagramı verilen Silindir sınıfını kodlayınız. PI sabit üye olacaktır. Varsayılan değeri, 3.14 olacaktır. Ama sınıfın kullanıcısı istediği sabit değeri atayabilecektir.

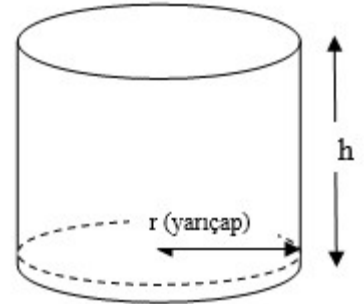
(b) Ayrıca, Silindir nesnesinin kullanıldığı bir uygulama yazınız. Uygulama çalıştırıldığında aşağıda gösterilen örnek çıktı alınabilecektir.

Silindir
- yükseklik: int - yarıcap: int - PI: float <constant>
+ setParameters(h: int, r: int) + yuzeyAlaniHesapla(): double + hacimHesapla(): double

Yüzey alanı:

$$A = 2\pi rh + 2\pi r^2$$

$$\text{Hacim: } H = \pi r^2 h$$



Örnek çıktı: (Koyu karakterle gösterilenler kullanıcı tarafından girilecek değerlerdir.)

```
Yukseklık (h): 5↵
Taban yarıcapı (r): 3↵
Yuzey alanı (A): 150.756
Hacmi (H): 141.372
```

Cevap:

Başarılar...

```
(a)
class Silindir {
public:
    Silindir(int y=0,int ycap=0,float pi=3.14)
        :PI(pi) {
        yukseklik = y;
        yaricap = ycap;
    }
    void setParameters(int h, int r) {
        yukseklik = h;
        yaricap = r;
    }
    double yuzeyAlaniHesapla() {
        double area;
        area = 2 * PI*yaricap*yukseklik +
              2 * PI* yaricap *yaricap;
        return area;
    }
    double hacimHesapla() {
        double volume;
        volume=PI*yaricap*yaricap*yukseklik;
        return volume;
    }
private:
    int yukseklik;
    int yaricap;
    const float PI;
};
```

```
(b)
#include <iostream>
using namespace std;

int main() {
    int h, r;
    cout << "Yukseklık (h) : ";
    cin >> h;
    cout<< "Taban yarıcapı (r) :";
    cin >> r;
    Silindir s(h,r) ;
    cout << "Yuzey alanı (A) : "
         << s.yuzeyAlaniHesapla()
         << endl;
    cout << "Hacmi (H) : "
         << s.hacimHesapla()
         << endl;
    return 0;
}
```