

152113017: NESNE TABANLI PROGRAMLAMA I
ARA SINAV (12.11.2015)

Kitap ve notlar kapalı

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

S.1	S.2	S.3	S.4	Toplam
20	30	20	30	100

Adı-Soyadı: Cevaplar

Numara: Cevaplar

S.1 (a) (20 points, each 4 points) Çıktıyı yaz. Kısmi puan verilmez.

(1) Programın çıktısı nedir?

Cevap: 0 1 2 3 2 2

(2) Programın çıktısı nedir?

Cevap: <Merhaba> 20 <Hello> 25

(3) Programın çıktısı nedir?

Cevap: 5 7 / 2 2 / 2 3 /

(4) Programın çıktısı nedir?

Cevap: 4 1 6

(5) Programın çıktısı nedir?

Cevap: 395

```
(3)
int main(){
    int *p1 = new int[2], *p2 = new int(7);
    p1[0] = 5; p1[1]=2;
    cout << p1[0] <<" "<< *p2 <<" / " ;
    *p2 = p1[1];
    cout << p1[1] <<" "<< *p2 <<" / " ;
    p1++;
    (*p2)++;
    cout << *p1 <<" "<< *p2 <<" / " ;
    return 0;
}
```

```
(4)
void kaos1(int b, int a, int &c){
    a+=1;b+=1;c=a+b;
}
void kaos2(int b, int &a){
    a+=2;b+=2;
}
void kaos3(int &b, int a){
    a+=3;b+=3;
}
int main(){
    int a,b,c;
    a=b=c=1;
    kaos1(a,b,c);
    kaos2(b,c);
    kaos3(a,c);
    cout << a <<" "<< b <<" "<< c <<
endl;
    return 0;
}
```

```
(1)
int x=0;
int main(){
    cout << x <<" ";
    int x=1;
    {
        cout << x <<" ";
        x = 2;
        {
            cout << x <<" ";
            int x = 3;
            cout << x <<" ";
        }
        cout << x <<" ";
    }
    cout << x <<" ";
    return 0;
}
```

```
(2)
namespace Mehmet{
    void message(){cout<<"<Merhaba> ";}
    int howOld(){return 25;}
}
namespace John{
    void message(){cout<<"<Hello> ";}
    int howOld(){return 20;}
}
using namespace Mehmet;
int main(){
    using John::howOld;
    message();
    cout<<howOld()<<" ";
    John::message();
    cout<<Mehmet::howOld()<<" ";
    return 0;
}
```

```
(5)
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=1,y=2;
    double z=5.0;
    f(x,y);
    f(x,y,6);
    f(x,z);
    return 0;
}
```

S.2 (30p.) Aşağıdaki programı inceleyiniz. Programı nesne tabanlı forma dönüştürünüz. **(i)** Termometre sınıfı kodlanacaktır. Sınıf kodlamasında, veri saklama (data hiding) (8puan), sabit erişim fonksiyonları (const accessor function) (7puan), uygun kurucu fonksiyon (constructor function) (7puan) kuralına uygun olarak kodlanmalıdır. **(ii)** main programı, yeni oluşturduğunuz sınıf nesnelerini kullanacak şekilde yeniden yazınız(8puan).

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

struct Termometre{
    float deger;
    string birim;
};

void goster(const Termometre &t){
    cout << "Sicaklik : "
         << t.deger << " "
         << t.birim << endl;
}

bool esitlik(const Termometre &t1
            ,const Termometre &t2){
    if (t1.birim != t2.birim)
        return false;
    if (t1.deger != t2.deger)
        return false;
    return true;
}

int main(){
    Termometre dis={30,"Santigrat"};
    Termometre oda={25,"Santigrat"};

    goster(dis);
    goster(oda);

    if (esitlik(dis, oda))
        cout << "Esit sicakliklar"
              << endl;
    else
        cout << "farkli sicaklik"
              << endl;

    return 0;
}
```

Cevap:

(i)

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

class Termometre {
    float deger;
    string birim;
public:
    Termometre(float d = 0, string b="Santigrat")
        :deger(d), birim(b) {}
    void goster() const {
        cout << "Sicaklik : "
             << deger << " "
             << birim << endl;
    }
    bool esitlik(const Termometre &t) const {
        if (birim != t.birim)
            return false;
        if (deger != t.deger)
            return false;
        return true;
    }
};
```

veri saklama (data hiding)
(8puan)

Uygun kurucu fonksiyon
(constructor function) (7puan)

Sabit erişim fonksiyonları (const
accessor function) (7puan)

(ii) (8 puan)

```
int main() {
    Termometre dis( 30,"Santigrat");
    Termometre oda( 25,"Santigrat");

    dis.goster();
    oda.goster();

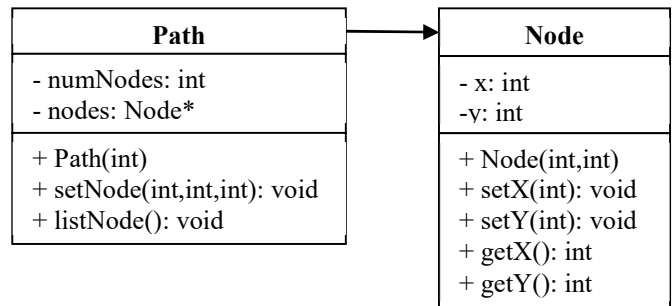
    if (dis.esitlik(oda))
        cout << "Esit sicakliklar"
              << endl;
    else
        cout << "farkli sicaklik"
              << endl;

    return 0;
}
```

S.3 (20p.) Aşağıdaki programı inceleyiniz.

```
class Node{
public:
    Node(int xx=0,int yy=0):x(xx),y(yy){}
    void setX(int x){this->x=x;}
    void setY(int y){this->y=y;}
    int getX(){return x;}
    int getY(){return y;}
private:
    int x;
    int y;
};

class Path{
public:
    Path(int num=0){numNodes=num; nodes=new Node[num];}
    void setNode(int index,int x,int y){
        nodes[index].setX(x);
        nodes[index].setY(y);
    }
    void listNodes(){
        for(int i=0;i<numNodes;i++)
            cout<<"node "<<i<<" : "<<nodes[i].getX()<<" , "<<nodes[i].getY()<<endl;
```



```

    }
private:
    Node *nodes;
    int numNodes;
};
int main() {
    Path myPath(1);
    for(int i=0;i<1;i++)
        myPath.setNode(i,i+1,i+2);
    myPath.listNodes();

    Path yourPath=myPath;
    yourPath.listNodes();

    myPath.setNode(0,7,7);
    myPath.listNodes();
    yourPath.listNodes();
    return 0;
}

```

a) (7p) Program derlenip, çalıştırıldığında ekran çıktısı ne olur?

b) (10p) myPath ve yourPath nesneleri ile ilgili bir problem vardır. Bu problem açıklayınız. Bu problem çözmek için Path sınıfına nasıl bir fonksiyon eklemelidir. Bu fonksiyon kodlarını yazınız. (İpucu: yapıcı “constructor” fonksiyon tiplerini düşünün.) c) (3p) Değişiklik sonrası, ekrana çıktısı ne olacaktır?

Cevap:

(a)

```

node 0 : 1, 2
node 0 : 1, 2
node 0 : 7, 7
node 0 : 7, 7

```

c)

```

node 0 : 1, 2
node 0 : 1, 2
node 0 : 7, 7
node 0 : 1, 2

```

(b) Satır “Path yourPath=myPath;” Path sınıfının copy constructor fonksiyonunu çağırır. Varsayılan(default) copy constructor fonksiyonu basit olarak değişken değerlerini kopyalar. Eğer sınıf içerisinde bir pointer varsa, sadece pointer tarafından tutulan adres kopyalanır. Halbuki, sadece adreste bulunan değer kopyalanması istenir. Bu problemi çözmek için, aşağıdaki gibi copy constructor fonksiyonunu sınıfın üye fonksiyonu olarak eklememiz gerekir:

```

Path(const Path &path) {
    this->numNodes=path.numNodes;
    this->nodes=new Node[path.numNodes];
    for(int i=0;i<this->numNodes;i++) {
        this->nodes[i].setX(path.nodes[i].getX());
        this->nodes[i].setY(path.nodes[i].getY());
    }
}

```

S.4 (30p) Bir kola makinesi programı yazalım. Programda, KolaMakinesi isimli aşağıdaki bileşenlerden oluşan bir sınıf tanımlayın:

Veri Üyeleri: Kolanın birim fiyatı (fiyat) ve makinadaki kalan kola sayısı (kolaSayisi)

Üye fonksiyonlar:

- Kurucu fonksiyon (constructor): birim fiyat ve kola sayısını parametre olarak alır. Parametreler için varsayılan değerler, birim fiyat 2,5 lira, kola sayısı 100 adettir.
 - kolaYukleme(): Makinaya yüklenen kola sayısını parametre alır. Makinadaki kola sayısı üzerine ekler.
 - kolaVer(): kullanıcıya birim fiyatı gösterir. Kullanıcıdan, atılan para miktarını girmesini bekler. Kullanıcı yeterli para girdi ise ve makinada kola varsa kullanıcıya bir kola verir (ekranda kolanın verildiğini gösterir) ve para üstünü hesaplayıp ekran gösterir.
 - göster(): Kola birim fiyatını ve makinada kalan kola sayısını ekranda gösterir.
- Sınıf kodlamasını yazdıktan sonra, main() fonksiyonunda bir kola makinesi nesnesi yaratın. Bir kullanıcının kola alması durumunda aşağıdaki örnek çıktının oluşmasını sağlayın.

Örnek Çıktı:

```

Kola Birim Fiyatı: 2,5 lira.
Para Atın: 3 ↵
Kolanız veriliyor...
Para üstünüz: 50 kuruş
Son Durum:
    Kola Birim Fiyatı : 2.5 lira.
    Kalan Kola Sayısı : 99 adet.

```

Cevap: KolaMakinesi.h :

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

class KolaMakinesi{
private:
    float fiyat;
    int kolaSayisi;
public:
    KolaMakinesi(float f = 2.5, int sayi = 100) :fiyat(f),kolaSayisi(sayi){}
    void kolaYukleme(int sayi){
        if(sayi>0)
            kolaSayisi += sayi;
    }
    void kolaVer();
    void goster() const;
};
```

KolaMakinesi.cpp :

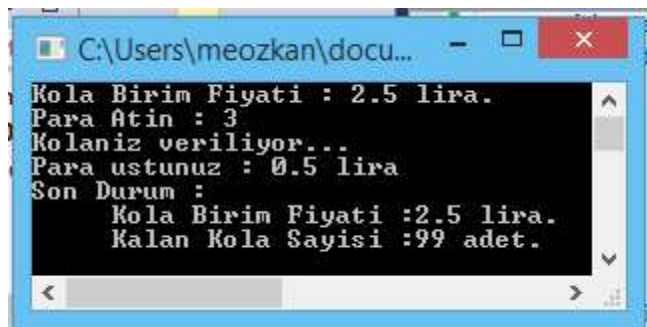
```
void KolaMakinesi::kolaYukleme(int sayi){
    if(sayi>0)
        kolaSayisi += sayi;
}
void KolaMakinesi::kolaVer(){
    float para;
    cout << "Kola Birim Fiyati : " << fiyat << " lira." << endl;
    cout << "Para Atin : ";
    cin >> para;
    if (para >= fiyat){
        kolaSayisi--;
        cout << "Kolaniz veriliyor..." << endl;
        cout << "Para ustunuz : " << (para - fiyat)<<" lira"<< endl;
    }
}
void KolaMakinesi::goster() const{
    cout << "Son Durum : " << endl
        << "      Kola Birim Fiyati : " << fiyat << " lira." << endl
        << "      Kalan Kola Sayisi : " <<kolaSayisi<< " adet."<<endl;
}
```

KolaMakinesiUygulama.cpp :

```
int main(){
    KolaMakinesi makine;

    makine.kolaVer();
    makine.goster();

    return 0;
}
```



Good lucks...