

#No:

Name Surname/Ad Soyad:

o 1. öğr.

o 2. öğr.

Object Oriented Programming / Final Exam - Nesne Yönelimli Programlama Final Sınavı

Q1/S1: You are asked to define a pure virtual class named Shape with **width** and **height** values. This class has a pure virtual **getArea ()** function. **Rectangle** and **Ellipse** are derived from this class. The main function of the program is given below. Answer the required questions accordingly. (the ellipse area is calculated as **(horizontal radius * vertical radius * pi)**)- Sizden **height** ve **width** değerlerine sahip **Shape** isminde saf sanal bir sınıf tanımlamanız istenmektedir. Bu sınıfın saf sanal **getArea()** fonksiyonu bulunmaktadır. Bu sınıftan **Rectangle** ve **Ellipse** sınıfı türetilmektedir. Programın main içeriği aşağıda verilmektedir. Buna göre istenen soruları cevaplandırınız (ellipse alanı **(yatay yarıçap*dikey yarıçap*pi)** şeklinde hesaplanmaktadır). (37p)

```
#include<iostream>
using namespace std;
#define PI 3.0
.....
int main(){
    Shape *ptr[5];
    Rectangle r1(3.0,5.0);
    Rectangle r2(3.5,4.0);
    Ellipse e1(4.0,6.0);
    Ellipse e2(4.0,7.0);
    ptr[0]=&r1; ptr[1]=&r2; ptr[2]=&e1; ptr[3]=&e2;
    double area = getTotalArea(ptr,4);
    cout<<"Total Area: "<<area<<endl;
    return 0;
}
```

a) Write the declarations of **Shape**, **Rectangle** and **Ellipse** classes. The **width** and **height** values should be sent to the constructor as parameters and the default values should be 0. - **Shape** saf sanal sınıfını ve bu sınıftan türeyen **Rectangle** ve **Ellipse** sınıflarının tanımlarını yazınız. **genişlik** ve **yükseklik** değerleri yapıcı fonksiyona parametre olarak gönderilmeli ve varsayılan değerleri 0 olmalıdır. (9p)

```
class Shape{
    protected:
        double height; // 1p
        double width;
    public:
        Shape(double _height=0.,double _width=0.); // 2p
        virtual double getArea() = 0; // 2p
};
class Rectangle : public Shape{
    public:
        Rectangle(double _height=0.,double _width=0.); (2p)
        double getArea();
};
class Ellipse : public Shape{
    public:
        Ellipse(double _height=0.,double _width=0.); (2p)
        double getArea();
};
```

b) Write the constructors of the **Shape**, **Rectangle** and **Ellipse** classes. When assigning initial values to the Rectangle and Ellipse constructors, use the initialization method by passing parameters to the constructor function of the Shape class. - **Shape**, **Rectangle** ve **Ellipse** sınıflarının yapıcı fonksiyonlarının kodunu yazınız. Rectangle ve Ellipse yapıcı fonksiyonlarında ilk değer atamaları yaparken Shape sınıfının yapıcı fonksiyonuna parametre aktararak ilk değer verme yöntemini kullanınız. (10p)

```
Shape::Shape(double _height, double _width){ // 2p
    height = _height;
    width = _width;
}
```

```
Rectangle::Rectangle(double _height, double _width)
    :Shape(_height, _width){ // 4p
}
Ellipse::Ellipse(double _height, double _width)
    :Shape(_height, _width){ // 4p
}
```

c) Implement the **getArea()** functions of the Rectangle and Ellipse classes. - Rectangle ve Ellipse sınıflarının **getArea()** fonksiyonlarını gerçekleştiriniz. (8p)

```
double Rectangle::getArea(){
    return height * width;
}
double Ellipse::getArea(){
    return (height * width * PI)/4;
} // her boşluk 2p
```

d) Fill in the spaces left in the **getTotalArea (...)** function, which takes a Shape pointer array and the number of object pointer numbers as a parameter and returns the total area of the shapes back. - Shape türünden bir pointer dizisi ve bu dizide bulunan nesne pointer sayısını parametre olarak alan ve geriye şekillerin toplam alanını döndüren **getTotalArea(...)** fonksiyonunda boş bırakılan yerleri doldurunuz. (6p)

```
double getTotalArea( Shape *_ptr[],int _n ){
    double sum = 0;
    for(int i=0;i<_n;i++){
        sum += _ptr[i]->getArea();
    }
    return sum;
} // her boşluk 2p
```

e) Write the screen output according to the main content given above. - Yukarıda verilen main içeriğine göre ekran çıktısını yazınız. (4p)

```
Total Area: 68 // 4p
```

Q2/S2: You will be asked to write a class that calculates the 11-digit TC ID number from a 9-digit number given as a string. The algorithm is as follows: - string şeklinde verilen 9 haneli kütük numarası kullanılarak 11 haneli TC kimlik numarasını hesaplayan bir sınıf yazmanız istenmektedir. Algoritma şu şekildedir:

- The first 9 digits of the TC ID number are the same as the 9-digit number. - TC kimlik numarasının ilk 9 hanesi kütük numarası ile aynıdır.
- The 10th digit of the TC ID number is the number in the first digit of the result obtained by subtracting the sum of the 2, 4, 6 and 8 digits from the 7 times the sum of the digits in the 1, 3, 5, 7 and 9 digits. - TC kimlik numarasının 10. hanesi 1, 3, 5, 7 ve 9. hanelerdeki rakamların toplamının 7 katından 2, 4, 6 ve 8. hanelerin toplamı çıkarılarak elde edilen sonucun birer basamağındaki rakamdır.
- 11th digit is the first digit of the sum of the first 10 digits. - 11. hane ilk 10 hanedeki rakamların toplamının birer basamağındaki rakamdır.

Main content to be used in the program is as follows. - Programda kullanılacak main içeriği aşağıdaki gibidir. (33p)

```
int main(){
    CitizenID tc,tc2("123456789");
    cout<<"TCKN of Atatürk: "<<tc.getCitizenID()<<endl;
    cout<<"TCKN 2: "<< tc2.getCitizenID() <<endl;
    return 0;
}
```

Object Oriented Programming / Final Exam - Nesne Yönelimli Programlama Final Sınavı

a) Define a class named CitizenID with the following properties (make sure the function prototypes are correct). - Aşağıdaki özelliklere sahip CitizenID isimli sınıfı tanımlayınız (fonksiyon prototiplerinin doğru olmasına dikkat ediniz). (6p)

- must have the 9 digit ID variable of string as private, - private erişim düzenleyici ile etiketli string türünden 9 haneli ID değişkenine sahip olmalıdır,
- constructor should take a string and default value should be "100000001" which is the id of M. Kemal ATATÜRK, - yapıcı fonksiyon bir string almalı ve varsayılan değeri M. Kemal ATATÜRK'ün kütük numarası "100000001" olmalıdır,
- to set the ID variable, you must have a setter function named setID (...), - ID değişkenini setlemek için setID(...) isimli bir setter fonksiyon olmalıdır,
- getCitizenID(...) must be a getter function that returns the 11-digit TC ID number as a string. - getCitizenID(...) isminde 11 haneli TC kimlik numarasını string olarak geri döndüren getter fonksiyon olmalıdır.

```
class CitizenID{
public:
    CitizenID(string _id = "100000001"); // 2p
    void setFamilyID(string _id); // 1p
    string getCitizenID(); // 1p
private:
    string familyID; // private olması 2p  };

```

b) Implement the constructor and setID (...) of the class. - Sınıfın yapıcı ve setID(...) fonksiyonunu gerçekleştiriniz. (8p)

```
CitizenID::CitizenID( string _id ){
    familyID = _id; }
void CitizenID::setID( string _id ){
    familyID = _id;
} // her boşluk 2p

```

c) In the function getCitizenID (...), which returns the 11-digit TC ID number, fill in the blanks according to the above algorithm. - Geriye 11 haneli TC kimlik numarasını döndüren getCitizenID(...) isimli fonksiyon kodunda yukarıdaki algoritmaya uygun olacak şekilde boş bırakılan yerleri doldurunuz. (13p)

```
string CitizenID::getCitizenID(){ // 1p
    short int t[11];
    for(int i=0;i<9;i++){
        t[i] =familyID[i] - '0'; // 3p
    }
    t[9] = ( 7*(t[0]+t[2]+t[4]+t[6]+t[8]) // 2p+2p
            -(t[1]+t[3]+t[5]+t[7]) )%10;
    int sum = 0;
    for(int i=0;i<10;i++){
        sum += t[i]; // 2p
    }
    t[10] = sum % 10;
    char rt[11];
    for(int i=0;i<11;i++){
        rt[i] = t[i] + '0'; // 3p
    }
    return string(rt); }

```

d) Write the screen output when the main function given above. - Yukarıda verilen main fonksiyonu çalıştırıldığında ekran çıktısının ne olacağını yazınız. (6p)

TCKN of Atatürk: 10000000146
TCKN 2: 12345678950

Q3/S3: You are required to implement a class hierarchy which lets programmer to process the objects as given in the example below. - Sizden nesneleri aşağıdaki örnekte verildiği şekilde işleyen bir sınıf hiyerarşisi kodlamanız istenmektedir. (30p)

```
void write_buffer_items(bufferType &buffer, int item){
    buffer.writeItem(item);
}

int main()
{
    int stack_items[] = {1,2,3,4,5};
    int fifo_items[] = {6,7,8,9,10};
    stack s(sizeof(stack_items)/sizeof(int));
    fifo f(sizeof(fifo_items)/sizeof(int));
    for(unsigned int i=0; i<sizeof(stack_items)/sizeof(int); i++)
    {
        write_buffer_items(s, stack_items[i]);
    }
    for(unsigned int i=0; i<sizeof(fifo_items)/sizeof(int); i++)
    {
        write_buffer_items(f, fifo_items[i]);
    }
    return 0;
}

```

Please complete the class lists given below with the correct definitions - aşağıdaki sınıf listelerini doğru olarak doldurunuz.

class bufferType //5 points

```
{
public:
    bufferType();
    virtual ~bufferType();

```

protected:

```
};
class fifo : public bufferType
{
public:
    fifo();
    fifo(long long);
    virtual ~fifo();
    int writeItem(int);
protected:
    bool isBufferEmpty();
    bool isBufferFull();
    int getCurrentDepth();
private:
    int readIndex;
};
int fifo::writeItem(int itemIn){ //5 points

```

bool fifo::isBufferEmpty(){ //2 points

int fifo::getCurrentDepth(){ //5 points

class stack : public bufferType

```
{
public:
    stack();
    stack(long long);
    virtual ~stack();
    int writeItem(int);
    const int& operator[](int index)const;
protected:
    bool isBufferEmpty();
    bool isBufferFull();
private:

```

const int& stack::operator[](int index)const{ //3 points

bool stack::isBufferEmpty(){ //2 points

bool stack::isBufferFull(){ //3 points

Exam Rules - Sınav Kuralları

1. It is compulsory to follow the rules of the examination announced on the web site - Bölüm sayfasında ilan edilen sınav kurallarına uymak zorunludur.
2. Questions are associated with program outputs 2 and 3. - Sorular 2 ve 3 numaralı program çıktıları ile ilişkilidir