

ობიექტზე ორიენტირებული დაპროგრამება 1 (C++/C#-ის
ბაზაზე)(C++)(108850, 108851, 108852, 108853 ჯგ)
I სემესტრი. 2019-2020 წელი
(დასკვნითი გამოცდის ნიმუში)

Question 1

რომელი კონსტრუქციით შეიძლება იყოს კლასი განსაზღვრული:

Select one:

- a. კლასის_ტიპი კლასის_სახელი {კლასის_წევრების_სია}
- b. კლასის_სახელი {კლასის მონაცემები, კლასის მეთოდები}
- c. კლასის_სახელი {კლასის წევრების სია}
- d. კლასის_ტიპი {კლასის მონაცემები, კლასის ფუნქციები}

Question 2

რომელი ტერმინი გამოიყენება კლასში განსაზღვრული ფუნქციისთვის?

Select one:

- a. ცვლადი წევრი
- b. ფუნქცია წევრი
- c. კლასის ფუნქცია
- d. კლასიკური ფუნქცია

Question 3

რა მიიღება შემდეგი პროგრამის შესრულების შედეგად? (შედეგები გამოყავით მძიმით)

```
class Bix
```

```
{
```

```
    public:
```

```
        int x;
```

```
};
```

```
int _tmain(int argc, _TCHAR* argv[])
```

```
{ Bix *p = new Bix();
```

```
    (*p).x = 10;
```

```
    cout<< (*p).x << ", " << p->x << ", " ;
```

```
    p->x = 20;
```

```
    cout<< (*p).x << ", " << p->x ;
```

```
return 0;  
}
```

Answer:

Question 4

მოცემულ ფრაგმენტში კლასის რომელი ტიპის წვდომას მიეკუთვნება a ცვლადი?

```
class abc {  
int a;  
public: void F(){ a++; }  
};  
  
int _tmain(int argc, _TCHAR* argv[])  
{abc dd;  
dd.F();  
.....  
}
```

Select one:

- a. დაცული
- b. სამივე პასუხი სწორია
- c. ღია (საჯარო)
- d. დახურული (კერძო)

Question 5

რა შედეგს მოგვცემს შემდეგი პროგრამა?:

```
class IndiaBix  
{  
int val;  
public:  
void SetValue(char *str1, char *str2)  
{  
val = strstr(str1, str2);  
}  
void ShowValue()  
{  
cout<< val;  
}  
};
```

```
int _tmain(int argc, _TCHAR* argv[])
{
    IndiaBix objBix;
    objBix.SetValue((char*)"India", (char*)"Bix");
    objBix.ShowValue(); system("pause");
    return 0;
}
```

Answer:

Question 6

ფუნქციების გადატვირთვა ნიშნავს რამდენიმე ფუნქციის განსაზღვრას, რომელთაც აქვთ:

Select one:

- a. ერთნაირი სახელები და განსხვავებული პარამეტრები
- b. სხვადასხვა სახელები და სხვადასხვა პარამეტრები
- c. ერთნაირი სახელები და ერთნაირი პარამეტრები
- d. სხვადასხვა სახელები და ერთნაირი პარამეტრები

Question 7

ჩამოთვლილთაგან აირჩიეთ სწორი განმარტება:

Select one:

- a. კონსტრუქტორი არის კლასის მეთოდი, რომელიც ამარტივებს კლასის მონაცემ-ელემენტების ინიციალიზებას;
- b. კონსტრუქტორი არის მეთოდი, რომელიც გამოიყენება კლასის მხოლოდ დახურულ მონაცემებზე წვდომისათვის;
- c. კონსტრუქტორი არის მეთოდი, რომელიც გამოიყენება კლასის მხოლოდ დაცულ მონაცემებზე წვდომისათვის.
- d. კონსტრუქტორი არის კლასის მეთოდი, რომელიც გამოიყენება კლასის მხოლოდ ღია მონაცემებზე წვდომისათვის;

Question 8

მოცემულ ფრაგმენტში რომელია კონსტრუქტორი?:

```
class queue {
int q[100];
int sloc, rloc;
public:
queue ();
void qput(int i);
int qget ();
```

```
}
```

Select one:

- a. queue ();
- b. int q[100];
- c. void qput(int i);
- d. int qget ();

Question 9

```
class Bix
{
    int x, y;
public:
    Bix(int xx, int yy)
    {
        x = xx;
        y = yy;
    }
    void show()
    {
        cout<< x * y << endl;
    }
};

int _tmain(int argc, _TCHAR* argv[])
{
    Bix obj(10, 20);
    obj.show();system("pause");
    return 0;
}
```

Answer:

Question 10

ჩამოთვლილთაგან რომელია სწორი გამოთქმა?

Select one:

- a. არცერთი არ არის სწორი
- b. კონსტრუქტორი კონკრეტული ეგზემპლარისთვის არ გამოიძახება ერთჯერადად;
- c. კონსტრუქტორი კონკრეტული ეგზემპლარისთვის გამოიძახება მრავალჯერ;
- d. კონსტრუქტორი კონკრეტული ეგზემპლარისთვის გამოიძახება მხოლოდ ერთჯერ - მისი შექმნისას;

Question 11

რომელი გადატვირთული კონსტრუქტორია გამოყენებული მოცემულ ფრაგმენტში?

```
class _2Data
{
double data1;
double data2;
public:
_2Data();
_2Data(double initData1, double initData2);
void showData();
};
_2Data::_2Data(double initData1, double initData2)
{
data1 = initData1;
data2 = initData2;
}
_2Data::_2Data()
{
data1 = 0;
data2 = 0;
}
```

Select one:

- a. void showData()
- b. double data1 და double data2
- c. using namespace std;
- d. data1 = initData1 და data2 = initData2
- e. _2Data() და _2Data(double initData1, double initData2)

Question 12

რაში მდგომარეობს ოპერატორების გადატვირთვა?

Select one:

- a. ოპერატორის აზრის შეცვლა პროგრამის დასრულებისას
- b. ოპერატორის აზრის შეცვლა მთელი პროგრამისთვის
- c. ოპერატორის აზრის შეცვლა პროგრამის დასაწყისში
- d. ოპერატორის აზრის შეცვლაში განსაზღვრულ კლასთან მიმართებით

Question 13

მოცემულ ფრაგმენტში რომელია გადატვირთული ოპერატორი?

```
class string
{
public:
    string(char *);
    void operator +(char *);
    void operator -(char);
    void show_string(void);
private:
    char data[256] ;
};
```

Select one or more:

- a. void operator +(char *);
- b. void show_string(void);
- c. string(char *);
- d. void operator -(char);

Question 14

რომელი საკვანძო სიტყვით აღიწერება კლასის სტატიკური წევრი?:

Select one:

- a. public
- b. protected
- c. class
- d. static

Question 15

რა დაიბეჭდება მოცემული პროგრამის შესრულების შემდეგ?

```
class Box {
public:
    static int objectCount;
    Box(double l = 2.0, double b = 2.0, double h = 2.0) {
        length = l;
        breadth = b;
        height = h;
        objectCount++;
    }
};
```

```

double Volume() {
    return length * breadth * height;
}
static int getCount() {
    return objectCount;
}
private:
    double length;
    double breadth;
    double height;
};
int Box::objectCount = 0;

int _tmain(int argc, _TCHAR* argv[])
{
    Box Box1(3.5, 1.2, 1.5);
    Box Box2(2.5, 6.5, 5.0);
    Box Box3(8.3, 4.6, 7.8);
    cout << Box::getCount() << endl;

}

```

Answer:

Question 16

რა არის მოცემულ პროგრამაში შეცდომის მიზეზი?

```

class IndiaBix
{
    static int x;
public:
    static void SetData(int xx)
    {
        x = xx;
    }
    void Display()
    {
        cout << x ;
    }
};

```

```
int IndiaBix::x = 0;
int _tmain(int argc, _TCHAR* argv[])
{
    IndiaBix::SetData(33);
    IndiaBix::Display();
    return 0;
}
```

Select one or more:

- a. Display() ფუნქცია არ არის სტატიკური
- b. Display() ფუნქციის გამოძახება არასწორია
- c. SetData(33) ფუნქცია ვერ მოახდენს მნიშვნელობის გადაცემას
- d. SetData(33) ფუნქცია პარამეტრიანია

Question 17

თუ პროგრამა იყენებს მემკვიდრეობითობას, მაშინ ახალი კლასის წარმოქმნისთვის საჭიროა?

Select one:

- a. მემკვიდრე კლასი
- b. არ არის საჭირო საბაზო კლასი
- c. არ არის საჭირო მემკვიდრე კლასი
- d. საბაზო კლასი

Question 18

მოცემული კოდის მიხედვით რომელი კლასი არის საბაზო?

```
class manager : public employee
```

Select one:

- a. ორივე კლასი საბაზოა
- b. manager
- c. employee
- d. არცერთი კლასი არ არის საბაზო

Question 19

მოცემული ფრაგმენტის მიხედვით რომელი მინიჭებებია სწორი წარმოებულ კლასში?

```
class some {
public:
    int a_;
protected:
    int b_;
private:
    int c_;
};
```



```
.....  
class derived : public some {  
    derived() {  
        a_ = 0;  
        b_ = 0;  
        c_ = 0;  
    }  
};
```

Select one or more:

- a. a_=0
- b. c_=0
- c. სამივე სწორია
- d. b_=0

Question 20

რა დაიბეჭდება მოცემული პროგრამის შესრულების შემდეგ?:

```
class Shape {  
    public:  
        void setWidth(int w) {  
            width = w;  
        }  
        void setHeight(int h) {  
            height = h;  
        }  
    protected:  
        int width;  
        int height;  
};  
class Rectangle: public Shape {  
    public:  
        int getArea() {  
            return width * height;  
        }  
};  
  
int _tmain(int argc, _TCHAR* argv[])  
{  
    Rectangle Rect;
```

```
Rect.setWidth(7);
Rect.setHeight(5);
cout<<Rect. getArea();
.....
}
```

Answer:

Question 21

რომელია დაცული წვდომის სპეციფიკატორი:

Select one:

- a. სამივე
- b. protected
- c. public
- d. private

Question 22

მოცემულია პროგრამის ფრაგმენტი. void update() ფუნქციაში რომელი მინიჭებებია სწორი?:

```
class a
{
    private: int x;
    public: int y;
    protected: int z;
};

class b:public a
{
    public:
    void update()
    {
        x=100;
        y=100;
        z=100;
        cout<<y<<" "<<z;    }
};
```

Select one:

- a. y=100; z=100;
- b. სამივე: x=100; y=100; z=100;
- c. მხოლოდ y=100;

- d. მხოლოდ $x=100$;

Question 23

რა შემთხვევაში მიიღება მრავლობითი მემკვიდრეობითობა?:

Select one:

- a. როდესაც კლასი შეიქმნება ღია მემკვიდრეობით;
- b. როდესაც კლასი შეიქმნება რამდენიმე საბაზო კლასისგან;
- c. როდესაც კლასი შეიქმნება დაცული მემკვიდრეობითობით.
- d. როდესაც კლასი შეიქმნება ერთი საბაზო კლასისგან;

Question 24

მოცემული ფრაგმენტიდან გამომდინარე რომელი გამონათქვამია სწორი class C -სთან მიმართებით?

```
class A
{public:
A() {}
.....
};
class B
{public
B() {}
.....
};

class C: public A, public B
{
C(){}
.....
};
```

Select one or more:

- a. მიღებულია მრავლობითი მემკვიდრეობითობით;
- b. მიღებულია ერთი კლასის ბაზაზე;
- c. საჯარო მემკვიდრეობითობით მიღებული წარმოებული კლასია;
- d. არ არის წარმოებული კლასი.

Question 25

რას გამოიტანს შემდეგი პროგრამა?

```
class BaseClass1 {
public:
BaseClass1() {
cout << "BaseClass1 constructor." << endl;
```

```

    }
};
class BaseClass2 {
public:
    BaseClass2() {
        cout << "BaseClass2 constructor." << endl;
    }
};
class DerivedClass : public BaseClass1, public BaseClass2 {
public:
    DerivedClass() {
        cout << "DerivedClass constructor." << endl;
    }
};

int _tmain(int argc, _TCHAR* argv[])
{ DerivedClass dc;
}

```

Select one or more:

- a. BaseClass1 constructor
- b. არცერთს არ გამოიტანს
- c. BaseClass2 constructor
- d. DerivedClass constructor

Question 26

პროგრამის რომელ ნაწილში განისაზღვრება ვირტუალური ფუნქცია?

Select one:

- a. საბაზო კლასში
- b. წარმოებულ კლასში
- c. დახურულ კლასში
- d. მეგობარ კლასში

Question 27

რა შედეგს გამოიტანს მოცემული პროგრამა? (შედეგები გამოყავით მძიმით)

```

class X {
protected:
    int i;

```

```

public:
    void seti(int c) {i = c;}
    virtual void print() {cout << i<<",";}
};
class Y : public X {
public:
    void print() {cout << i<<endl;}
};

int _tmain(int argc, _TCHAR* argv[])
{
    X x;
    X *px=&x;
    Y y;
    x.seti(10);
    y.seti(15);
    px->print();
    px=&y;
    px->print();
    return 0;
}

```

Answer:

Question 28

ჩამოთვლილთაგან რომელია სწორი განმარტება მეგობარ ფუნქციასთან მიმართებით?

Select one:

- a. friend ფუნქცია არ არის კლასის წევრი, მაგრამ მას აქვს წვდომა კლასის მხოლოდ დაცულ წევრებთან
- b. friend ფუნქცია არის კლასის წევრი, მას აქვს წვდომა კლასის დახურულ და დაცულ წევრებთან
- c. friend ფუნქცია არის კლასის წევრი, მაგრამ მას არა აქვს წვდომა კლასის დახურულ და დაცულ წევრებთან
- d. friend ფუნქცია არ არის კლასის წევრი, მაგრამ მას აქვს წვდომა კლასის დახურულ და დაცულ წევრებთან

Question 29

მოცემული პროგრამიდან გამომდინარე, რომელი განმარტებაა მართებული?

```

class Point
{
    friend void ChangePrivate( Point & );
}

```

```

public:
    Point( void ) : m_i(0) {}
    void PrintPrivate( void ){cout << m_i << endl; }

private:
    int m_i;
};
void ChangePrivate ( Point &i ) { i.m_i++; }
int _tmain(int argc, _TCHAR* argv[])
{ Point sPoint;
  sPoint.PrintPrivate();
  ChangePrivate(sPoint);
  sPoint.PrintPrivate();
  system("pause");
  return 0;
}

```

Select one or more:

- a. ChangePrivate ფუნქციას აქვს წვდომა Point კლასის დახურულ მონაცემ-წევრზე
- b. კლასის მეგობარ ფუნქციას აქვს წვდომა ამ კლასის დახურულ მონაცემ-წევრზე
- c. კლასის მეგობარ ფუნქციას არა აქვს წვდომა ამ კლასის დახურულ მონაცემ-წევრზე
- d. კლასის მეგობარი ფუნქცია არასწორად არის განსაზღვრული

Question 30

რას გვამღევეს ისეთი შემთხვევა, როდესაც ერთ კლასში ცხადდება სხვა კლასი, როგორც მეგობარი?

Select one:

- a. კლასი მეგობარ კლასს არ აძლევს თავის მონაცემებზე წვდომის საშუალებას
- b. კლასი მეგობარ კლასს აძლევს თავის მხოლოდ დაცულ მონაცემებზე წვდომის საშუალებას
- c. კლასი მეგობარ კლასს აძლევს თავის მხოლოდ დახურულ მონაცემებზე წვდომის საშუალებას
- d. კლასი მეგობარ კლასს აძლევს თავის დახურულ და დაცულ მონაცემებზე წვდომის საშუალებას

Question 31

მოცემული ფრაგმენტიდან გამომდინარე რომელი განმარტებაა მართებული?

```

class AClass {
    friend class BClass;
private:
    double value;
public:
    AClass() { value = 3.14159; }
};

```

Select one or more:

- a. BClass კლასი AClass კლასის მეგობარია
- b. value მისაწვდომია AClass და BClass კლასების წევრებისთვის
- c. value მისაწვდომია მხოლოდ AClass კლასის წევრებისთვის
- d. value არ არის მისაწვდომი არცერთი კლასის წევრებისთვის