

ობიექტზე ორიენტირებული დაპროგრამება 1 (C++/C#-ის
ბაზაზე)(C++)(108650, 108651, 108652, 108653 ჯგ)
I სემესტრის 2017-2018 წელი
(დასკვნითი გამოცდის ნიმუში)

Question 1

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 2

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

Question 3

რა შედეგს მოგვცემს შემდეგი პროგრამა?: (შედეგები გამოყავით მძიმით)

```
class one
{
private:
    int a, b;
public:
    void more (int x, int y)
    { a=4,b=7;
      if (a>b) {a*=x;b*=y;} else {a+=x; b+=y;}
      cout<<a<<" ";
      cout<<b;
    }
};
int _tmain(int argc, _TCHAR* argv[])
{ one a;
  a.more(1,2); system("pause");}
```

Answer:

Question 4

ფუნქციების გადატვირთვა ნიშნავს რამდენიმე ფუნქციის განსაზღვრას, რომელთაც აქვთ:

Select one:

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

Question 5

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 6

რას გამოიტანს მოცემული პროგრამა?: (შედეგები გამოყავით მძიმით)

```
class AB  
{  
private:  
int a;  
int b;  
public:  
AB(int c, int d)  
{a=3;b=4;  
a+= c;  
b+= d;  
cout << a<<" ";  
cout << b;  
} };  
  
int _tmain(int argc, _TCHAR* argv[])  
{  
  
AB obj1(100, 200); system("pause");
```

```
}
```

Answer:

Question 7

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

Question 8

რა დაიბეჭდება მოცემული პროგრამის შესრულების შემდეგ? (შედეგები გამოყავით მძიმით)

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

```
int Box::objectCount = 0;
```

```
int _tmain(int argc, _TCHAR* argv[])
{
    Box Box1(3.3, 1.2, 1.5);
    Box Box2(8.5, 6.0, 2.0);
    cout << Box::objectCount<<endl;
    system("pause");
}
```

Answer:

Question 9

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. SetData(33) ფუნქცია პარამეტრიანია
- c. Display() ფუნქციის გამოძახება არასწორია
- d. SetData(33) ფუნქცია ვერ მოახდენს მნიშვნელობის გადაცემას

Question 10

მოცემული კოდის მიხედვით რომელი კლასი არის საბაზო?

```
class manager : public employee
```

Select one:

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

Question 11

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. b_=0
- d. სამივე სწორეა

Question 12

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 13

მოცემულია პროგრამის ფრაგმენტი. void update() ფუნქციაში რომელი მინიჭებუბია 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. მხოლოდ y=100;
- b. მხოლოდ x=100;
- c. სამივე: x=100; y=100; z=100;
- d. y=100; z=100;

Question 14

რა შემთხვევაში მიიღება მრავლობითი მემკვიდრეობითობა?:

Select one:

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

Question 15

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

Question 16

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

Question 17

პროგრამის რომელ ნაწილში განისაზღვრება ვირტუალური ფუნქცია?

Select one:

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

Question 18

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 19

ჩამოთვლილთაგან რომელია სწორი განმარტება მეგობარ ფუნქციასთან მიმართებით?

Select one:

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

Question 20

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

Question 21

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

Select one:

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

Question 22

მოცემული ფრაგმენტიდან გამომდინარე რომელი განმარტებაა მართებული?

```

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

```

Select one or more:

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