



מס' נבחן

שם הקורס: סדנה בתכנות מונחה עצמים ושפת ++C
קוד הקורס: 10216

הוראות לנבחן:

-חומר עזר שימושי לבחינה
 כל חומר עזר מודפס

-אין לכתוב בעפרון / עט מחיק
 -אין להשתמש בטלפון סלולארי
 -אין להשתמש במחשב אישי או נייד
 -אין להשתמש בדיסק און קי ו/או
 מכשיר מדיה אחר
 -אין להפריד את דפי שאלון הבחינה

בחינת סמסטר: קיץ

השנה: תשע"ח

מועד: 7C

28.9.18
 8:30

תאריך הבחינה:

שעת הבחינה

משך הבחינה: 180 דקות

**השאלון לא ייבדק בתום הבחינה ע"י
 המרצה**

מרצה: בוריס מורוז

מבנה הבחינה והנחיות לפתרון:

- במבחן יש 8 עמודים ובהן 10 שאלות, יש לענות על כול השאלות.
- עבור כל שאלה יש לבדוק האם התכנית עוברת קומפילציה. אם לא – יש לפרט מהי השגיאה, אם כן – יש לפרט מהו הפלט.
- **שימו לב:** שאלה שאינה עוברת קומפילציה עם נימוק שאינו נכון, אינה מזכה בנקודות.
- ניקוד כל שאלה הינו 10 נקודות

בהצלחה!

כל הזכויות שמורות ©. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכאנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה

שאלות 5-1 יתייחסו לקטע הקוד הבא:

```
#include "stdafx.h"
#include <iostream>
using namespace std;

template <class T, class K>
class Couple {
private:
    T objT;
    K objK;
public:
    Couple(int i):objT(i), objK(i) {
    }
    Couple(T &t, K &k) : objT(t), objK(k) {}

    ~Couple() {
        cout << "In Couple::~~Couple" << endl;
    }
    Couple& operator+(const T& other) {
        objT += other;
        return *this;
    }
    Couple& operator+(const K& other) {
        objK += other;
        return *this;
    }
    void show() const {
        objK.show();
        objT.show();
    }

    T& getT() { cout << "In getT" << endl; return objT; }
    K& getK() { cout << "In getK" << endl; return objK; }
};
```



```

class First {
    int value;
public:
    First(int x = 1) {
        value = x;
        cout << "Constructor First value = " << value << endl;
    }

    First(const First &other) {
        value = other.value;
        cout << "First copy constructor value = " << value << endl;
    }
    ~First() {
        cout << "Destructor First value = " << value << endl;
    }

    friend ostream& operator<<(ostream& os, const First& b);

    void show() const { cout << "in First value: " << value << endl; }

    First& operator+=(const First& other) {
        value += other.value;
        return *this;
    }

    First& operator++(int) {
        ++value;
        return *this;
    }
    int getValue() { return value; }
};

ostream& operator<<(ostream& os, const First& b) {
    return os << "Operator << in First b.value = " << b.value << endl;
}

```



```

class Second {
public:
    char *value;
    Second(int i = 9, char *data = "Jerusalem") {
        value = new char[i+1];
        strncpy(value, data, i); // Copy i chars from data to value
        value[i] = '\0';
        cout << "Constructor Second value = " << value << endl;
    }
    Second(const Second &other) {
        value = _strdup(other.value); // duplicates string
        cout << "Second copy constructor value = " << value << endl;
    }
    ~Second() {
        cout << "Destructor Second value = " << value << endl;
        delete [] value;
    }
    void show() const { cout << "in Second value: " << value << endl; }
};

ostream& operator<<(ostream& os, const Second& b) {
    return os << "Operator << in Second  b.value = " << b.value << endl;
}

```

שאלה 1:

```

void main()
{
    First f1, f2(20);
    Second s1, s2(30);
    Couple<First, Second> c(f1, s1);
}

```

שאלה 2:

```

void main2() {
    First f1(4), f2(f1);
    Couple<First, First> c(f1, f2);
}

```


שאלה 3:

```
void main() {  
    First f(20);  
    Second s(5, "Afeka");  
    Couple<First, Second> c(f, s);  
    c + f;  
}
```

שאלה 4:

```
void main4() {  
    First f(20);  
    Second s(5, "Afeka");  
    Couple<First, Second> c(f, s);  
    c + f + s;  
}
```

שאלה 5:

```
void main() {  
    First f1(20), f2(30);  
    Second s(5, "Afeka");  
    Couple<First, Second> c(f1, s);  
    c.getT() = f2;  
    c.show();  
}
```


שאלות 6-10 יתייחסו לקטע הקוד הבא:

```
#include <iostream>
#include <vector>
#include <iterator>
#include <list>
#include <algorithm>

using namespace std;
class Base
{
public:
    int x;
    Base(int x) : x(x) { cout << "Constructor Base x = " << x << endl; }

    virtual void func() { cout << "Base::func y = " << x << endl; }
    virtual ~Base() { cout << "Destructor Base x = " << x << endl; }
};

class Derived : public Base
{
private:
    int y;
public:

    Derived(int x, int y=7) : Base(x), y(y) { cout << "Constructor Derived\n"; }

    Derived & operator=(int value) {
        x = y = value;
        cout << "Derived operator =" << endl;
        return *this;
    }
    Derived & operator<<(int value) {
        x = value;
        y = value;
        cout << "Derived operator <<" << endl;
        return *this;
    }
    Derived & operator-(int value) {
        x -= value;
        y -= value;
        cout << "Derived operator-" << endl;
        return *this;
    }

    void func() { cout << "Derived::func y = " << y << " x = " << x << endl;}
    ~Derived() { cout << "Destructor Derived y = " << y << endl; }
};
```



```

class Container
{
    vector<Base *> data;
public:
    Container(Base& d) {
        data.push_back(&d);
    }
    Container & operator+(Base& d) {
        cout << "Container operator+ x = " << d.x << endl;
        data.push_back(&d);
        return *this;
    }
    Base * &operator[](int index) {
        return data[index];
    }
    ~Container() {
        cout << "Destructor Container" << endl;
    }
    void func() {
        cout << "Derived::func" << endl;
        vector<Base *>::iterator it;
        for (it = data.begin(); it != data.end(); ++it)
            (*it)->func();
    }
};

```

שאלה 6:

```

void main() {

    Base b(10);
    Derived d1(20, 30), d2 = d1;
    Container c(b);
}

```

שאלה 7:

```

Derived f(Base b) {
    Derived d(7);
    d - b.x;
    return d;
}
void main() {
    Base b(100);
    Derived d = f(b);
}

```


פתרון

1001



מס' נבחן

שם הקורס: סדנה בתכנות מונחה עצמים ושפת ++C
קוד הקורס: 10216

בחינת סמסטר: קיץ
השנה: תשע"ח
מועד: 70

תאריך הבחינה:
שעת הבחינה
משך הבחינה: 180 דקות

השאלון לא ייבדק בתום הבחינה ע"י
המרצה

מרצה: בוריס מורוז

הוראות לנבחן:
- חומר עזר שימושי לבחינה
כל חומר עזר מודפס

- אין לכתוב בעפרון / עט מחיק
- אין להשתמש בטלפון סלולארי
- אין להשתמש במחשב אישי או נייד
- אין להשתמש בדיסק און קי ו/או
מכשיר מדיה אחר
- אין להפריד את דפי שאלון הבחינה

פתרון

בהצלחה!

כל הזכויות שמורות ©. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן
מאגר מידע, בכל דרך שהיא, בין מכאנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא
מטופס הבחינה

שאלות 5-1 יתייחסו לקטע הקוד הבא:

```
#include "stdafx.h"
#include <iostream>
using namespace std;

template <class T, class K>
class Couple {
private:
    T objT;
    K objK;
public:
    Couple(int i):objT(i), objK(i) {
    }
    Couple(T &t, K &k) : objT(t), objK(k) {}

    ~Couple() {
        cout << "In Couple::~~Couple" << endl;
    }
    Couple& operator+(const T& other) {
        objT += other;
        return *this;
    }
    Couple& operator+(const K& other) {
        objK += other;
        return *this;
    }
    void show() const {
        objK.show();
        objT.show();
    }

    T& getT() { cout << "In getT" << endl; return objT; }
    K& getK() { cout << "In getK" << endl; return objK; }
};
```



```

class First {
    int value;
public:
    First(int x = 1) {
        value = x;
        cout << "Constructor First value = " << value << endl;
    }

    First(const First &other) {
        value = other.value;
        cout << "First copy constructor value = " << value << endl;
    }
    ~First() {
        cout << "Destructor First value = " << value << endl;
    }

    friend ostream& operator<<(ostream& os, const First& b);

    void show() const { cout << "in First value: " << value << endl; }

    First& operator+=(const First& other) {
        value += other.value;
        return *this;
    }

    First& operator++(int) {
        ++value;
        return *this;
    }
    int getValue() { return value; }
};

ostream& operator<<(ostream& os, const First& b) {
    return os << "Operator << in First b.value = " << b.value << endl;
}

```



```

class Second {
public:
    char *value;
    Second(int i = 9, char *data = "Jerusalem") {
        value = new char[i+1];
        strncpy(value, data, i); // Copy i chars from data to value
        value[i] = '\0';
        cout << "Constructor Second value = " << value << endl;
    }
    Second(const Second &other) {
        value = _strdup(other.value); // duplicates string
        cout << "Second copy constructor value = " << value << endl;
    }
    ~Second() {
        cout << "Destructor Second value = " << value << endl;
        delete [] value;
    }
    void show() const { cout << "in Second value: " << value << endl; }
};

ostream& operator<<(ostream& os, const Second& b) {
    return os << "Operator << in Second  b.value = " << b.value << endl;
}

```

שאלה 1:

```

void main()
{
    First f1, f2(20);
    Second s1, s2(30);
    Couple<First, Second> c(f1, s1);
}
Constructor First value = 1
Constructor First value = 20
Constructor Second value = Jerusalem
Constructor Second value = Jerusalem
First copy constructor value = 1
Second copy constructor value = Jerusalem
In Couple::~~Couple
Destructor Second value = Jerusalem
Destructor First value = 1
Destructor Second value = Jerusalem

```


Destructor Second value = Jerusalem
Destructor First value = 20
Destructor First value = 1

שאלה 2:

```
void main2() {  
    First f1(4), f2(f1);  
    Couple<First, First> c(f1, f2);  
}  
Couple<First, First> &Couple<First, First>::operator +(const T &)'  
member function already defined or declared
```

שאלה 3:

```
void main() {  
    First f(20);  
    Second s(5, "Afeka");  
    Couple<First, Second> c(f, s);  
    c + f;  
}  
  
Constructor First value = 20  
Constructor Second value = Afeka  
First copy constructor value = 20  
Second copy constructor value = Afeka  
In Couple::~~Couple  
Destructor Second value = Afeka  
Destructor First value = 40  
Destructor Second value = Afeka  
Destructor First value = 20
```

שאלה 4:

```
void main4() {  
    First f(20);  
    Second s(5, "Afeka");  
    Couple<First, Second> c(f, s);  
    c + f + s;  
}
```

binary '+=': 'Second' does not define this operator or a conversion
to a type acceptable to the predefined operator

שאלה 5:

```
void main() {
```



```

    First f1(20), f2(30);
    Second s(5, "Afeka");
    Couple<First, Second> c(f1, s);
    c.getT() = f2;
    c.show();
}

Constructor First value = 20
Constructor First value = 30
Constructor Second value = Afeka
First copy constructor value = 20
Second copy constructor value = Afeka
In getT
in Second value: Afeka
in First value: 30
In Couple::~~Couple
Destructor Second value = Afeka
Destructor First value = 30
Destructor Second value = Afeka
Destructor First value = 30
Destructor First value = 20

```

שאלות 6-10 יתייחסו לקטע הקוד הבא:

```

#include <iostream>
#include <vector>
#include <iterator>
#include <list>
#include <algorithm>

using namespace std;
class Base
{
public:
    int x;
    Base(int x) : x(x) { cout << "Constructor Base x = " << x << endl; }

    virtual void func() { cout << "Base::func y = " << x << endl; }
    virtual ~Base() { cout << "Destructor Base x = " << x << endl; }
};

class Derived : public Base
{

```



```
private:
    int y;
public:
```

```
    Derived(int x, int y=7) : Base(x), y(y) { cout << "Constructor Derived\n"; }

    Derived & operator=(int value) {
        x = y = value;
        cout << "Derived operator =" << endl;
        return *this;
    }
    Derived & operator<<(int value) {
        x = value;
        y = value;
        cout << "Derived operator <<" << endl;
        return *this;
    }
    Derived & operator-(int value) {
        x -= value;
        y -= value;
        cout << "Derived operator-" << endl;
        return *this;
    }

    void func() { cout << "Derived::func y = " << y << " x = " << x << endl;}

    ~Derived() { cout << "Destructor Derived y = " << y << endl; }
};
```

```
class Container
{
    vector<Base *> data;
public:
    Container(Base& d) {
        data.push_back(&d);
    }
    Container & operator+(Base& d) {
        cout << "Container operator+ x = " << d.x << endl;
        data.push_back(&d);
        return *this;
    }
    Base * &operator[](int index) {
        return data[index];
    }
    ~Container() {
        cout << "Destructor Container" << endl;
    }
}
```



```

void func() {
    cout << "Derived::func" << endl;
    vector<Base *>::iterator it;
    for (it = data.begin(); it != data.end(); ++it)
        (*it)->func();
}
};

```

שאלה 6:

```

void main() {
    Base b(10);
    Derived d1(20, 30), d2 = d1;
    Container c(b);
}
Constructor Base x = 10
Constructor Base x = 20
Constructor Derived
Destructor Container
Destructor Derived y = 30
Destructor Base x = 20
Destructor Derived y = 30
Destructor Base x = 20
Destructor Base x = 10

```

שאלה 7:

```

Derived f(Base b) {
    Derived d(7);
    d - b.x;
    return d;
}
void main() {
    Base b(100);
    Derived d = f(b);
}
Constructor Base x = 100
Constructor Base x = 7
Constructor Derived
Derived operator-
Destructor Derived y = -93
Destructor Base x = -93
Destructor Base x = 100
Destructor Derived y = -93
Destructor Base x = -93
Destructor Base x = 100

```

שאלה 8:

```

void main() {

```



```

    Container c(Base(3));
    Derived d(1);
    Base b(2);
    c[0] = c[1];
}
Constructor Base  x = 3
Destructor Base  x = 3
Constructor Base  x = 1
Constructor Derived
Constructor Base  x = 2
Container operator+ x = 2
Container operator+ x = 1
Destructor Base  x = 2
Destructor Derived y = 7
Destructor Base  x = 1
Destructor Container

```

שאלה 9:

```

void main() {
    Base b(10);
    Derived d(20, 30);
    Container c(b);
    c + d;
    c.func();
}
Constructor Base  x = 10
Constructor Base  x = 20
Constructor Derived
Container operator+ x = 20
Derived::func
Base::func y = 10
Derived::func y = 30 x = 20
Destructor Container
Destructor Derived y = 30
Destructor Base  x = 20
Destructor Base  x = 10

```

שאלה 10:

```

Base *func(int i) {
    cout << "in f(int) i = " << i << endl;
    return new Base(i);
}

```

```

void main() {
    vector<int> x;
}

```



```

vector<Base *> b;

x.push_back(3);
x.push_back(5);
x.push_back(7);

transform (x.begin(), x.end(), back_inserter(b), func);
}
in f(int)  i = 3
Constructor Base  x = 3
in f(int)  i = 5
Constructor Base  x = 5
in f(int)  i = 7
Constructor Base  x = 7

```

בהצלחה!

כל הזכויות שמורות ©. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכאנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה