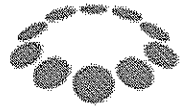


102
101

0600603025

המסלול האקדמי המכללה למינהל

ביה"ס למדעי המחשב



המסלול האקדמי
המכללה למינהל

ת.ז. הסטודנט: _____

מספר חדר: _____

מספר נבחן: _____

מספר אסמכתא: _____

ברקוד נבחן

מבחן בקורס: תכנות מונחה עצמים

תאריך הבחינה: 14.07.13

שנת הלימודים: תשע"ג, סמסטר: ב', מועד: א'

משך הבחינה: 3.5 שעות

שם המתרגל/ים:

חיים שפיר

שי צוויג

שם המרצה/ים:

אליהו חלסצ'י

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות הכולל בבחינה: 6. כאן ניתן להזין מלל חופשי, אם לא מחק/י.

משקל כל שאלה: בצמוד לכל שאלה

הוראות לנבחן:

- אסור השימוש בכל חומר עזר
- יש לענות בגוף השאלון.
- נדרש להחזיר את השאלון.
- לא מצורף נספח לבחינה
- מחברת טיוטה: כן
- מחברת נפרדת לכל שאלה: לא
-

בהצלחה!!

שאלה 1 – Constructors (10 נקודות)

נתונות ההגדרות של המחלקות הבאות :

```
class A{
public:
    int x;
    A(){
        x=0;
        cout<<"A Ctor"<<endl;
    }
};

class B: public A{
    A a;
public:
    B(){
        cout<<"B Ctor"<<endl;
    }
};

class C: public B{
    B b;
public:
    C(int x){
        A::x=x;
        cout<<"C Ctor"<<endl;
    }
};

int main(){
    C c(5);
    cout<<sizeof(c)<<endl;
}
```

א. מהו הפלט של התוכנית? (5 נק')

ב. סרטט/י את היצוג בזיכרון של אובייקט C שיצרנו ב main (5 נק')

(יש לסרטט אובייקט כמלבן, עם חלוקה למרכיביו השונים וייצוג ה data members שבתוכם וערכיהם)

תשובה:

א.

```
A Ctor
A Ctor
B Ctor
A Ctor
A Ctor
B Ctor
C Ctor
16
```

ב.

A::	int x=5				
B::	A a= int x=0				
C::	B b= <table> <tr> <td>A::</td><td>int x =0</td></tr> <tr> <td>B::</td><td>A a= int x=0</td></tr> </table>	A::	int x =0	B::	A a= int x=0
A::	int x =0				
B::	A a= int x=0				

שאלה 2: ירושה ופולימורפיזם (30 נקודות)

נתונה מחלקה Date ובה התכונות יום, חודש ושנה, בנאי ופעולות גישה.

כמו כן נתונה המחלקה "רכב דו-גלגלי" וממנה יורשות שתי המחלקות אופנוע ואופניים.

להלן קוד חלקי של המחלקות:

```
class Date
{
    int m_day,m_month,m_year;

public:
    Date(int day,int month, int year){
        m_day=day;
        m_month=month;
        m_year=year;
    }
}
```

```

};

class TwoWheelVehicle
{
    protected:
        char* model;           //דגם
        char* color;           //צבע
        Double maxSpeed; //מהירות מקסימלית
        Date manDate;          //תאריך ייצור

        // setters & getters...
        . . .
        . . .
};

class Bicycle : public TwoWheelVehicle
{
    bool helpWheels; //האם יש גלגלי עזר או לא
    // setters & getters...
    . . .
    . . .
};

Class MotorCycle: public TwoWheelVehicle
{
    Long licenseNum; //מספר רישוי

    . . .
    . . .
};

```

חלק א (12 נק')

- הגדירו בנאי למחלקה TwoWheelVehicle אשר יעדכן את כל התכונות שלה.
- הגדירו בנאי למחלקה Bicycle אשר יעדכן את כל התכונות שלה.
- נתונה תכנית. השלימו את התכנית כך שתייצר שני זוגות אופניים לפי המפרט הבא:
 זוג 1: דגם BMX ללא גלגלי עזר, מהירות מקסימלית 90 קמ"ש, תאריך ייצור 25/3/2013
 בצבע אדום (Red)
 זוג 2: דגם BMX ללא גלגלי עזר, מהירות מקסימלית 90 קמ"ש, תאריך ייצור 27/4/2013
 בצבע כחול (Blue).
 לאחר הייצור התכנית תעדכן את המהירות המקסימלית של הזוג הכחול להיות 100 קמ"ש.

```

int main()
{
    . . .
    . . .
    Bicycle b1 . . . ;
    Bicycle b2 . . . ;

    . . .
    . . .
}

```

חלק ב (18 נק')

נתונה המחלקה Store ובה מערך של כלי רכב דו גלגליים ובו אופניים ואופנועים:

```
private:
```

```
TwoWheelVehicle arr**;  
int arr_size; // number of elements in the above arr
```

(הניחו שהמשתנים הללו מאותחלים כראוי)

המשטרה רוצה לגייס לשורותיה אופניים והיא מעוניינת לדעת כמה זוגות אופניים בעלי מהירות מקסימלית גבוהה ביותר קיימים. (בהשוואה לאופניים אחרים בלבד ולא לאופנועים כמובן). אופני המשטרה הם ללא גלגלי עזר. כתבו מתודה בתוך המחלקה Store שתחזיר כמה זוגות אופניים ללא גלגלי עזר בעלי מהירות מקסימלית גדולה ביותר קיימים במערך. לדוגמה אם מהירויות האופניים ללא גלגלי עזר במערך הן: 100, 200, 150, 125, 200, 190, 100, 200 הערך המוחזר יהיה 3 כיוון שישנם 3 זוגות אופניים ללא גלגלי עזר שמהירותם היא המקסימלית (200). עליכם לבצע את הפעולה בצורה הכי יעילה האפשרית לדעתכם.

פתרון:

חלק א':

```
TowWheelVehicle(const char* amodel, const char* acolor, double amaxSpeed, Date  
amanDate) : mandate(amanDate) {  
    // initialization line for Date object using default copy CTOR  
    model=new char[strlen(amodel)+1];  
    strcpy(model,amodel);  
    color=new char[strlen(acolor)+1];  
    strcpy(color,acolor);  
    maxSpeed=amaxSpeed;  
}  
  
Bicycle(const char* amodel, const char* acolor, double amaxSpeed, Date amanDate, bool  
ahelpWheels) : TowWheelVehicle(amodel, acolor, amaxSpeed, amanDate) {  
    helpWheels=ahelpWheels;  
}
```

```
int main(){
```

```

Bicycle b1("BMX", "Red", 90, Date(25, 3, 2013), false);
Bicycle b2("BMX", "Blue", 90, Date(27, 4, 2013), false);
b2.setMaxSpeed(100);
}

```

חלק ב'

```

int countMaxSpeedBicycle(){
    int count=0, maxSpeed=0;
    for(int i=0; i<arr_size; i++){
        if (typeid(*arr[i]) == typeid(Bicycle)){ // this is a bicycle
            if( !((Bicycle*)arr[i])->hasHelpWheels() ){ // checks if helpWheels!=true
                if(arr[i]->getMaxSpeed()==maxSpeed)
                    count++;
                if(arr[i]->getMaxSpeed()>maxSpeed){
                    maxSpeed= arr[i]->getMaxSpeed(); // update max speed
                    count=1; // start a recount from now own
                }
            }
        }
    }
    return count;
}

```

שאלה 3: Const – Operator Overloading – (20 מקודות)

נתונה המחלקה הבאה המייצגת מחרוזת

```

class MyString{
    char* s;
public:
    MyString(const char* str){
        s = new char[strlen(str)+1];
        strcpy(s, str);
    }
    const char* getS() const {return s;}
};

int main(){

```

```

const MyString a("hello "),b("world!");
MyString c("something");
c=a+b;
cout<<c.getS()<<endl; // hello world!
}

```

ממשו את האופרטורים הדרושים לכך שהקוד ב main יעבוד, כאשר את אופרטור השרשור (+) יש לממש כפונקציה גלובאלית ולא כמתודה. (שימו לב לגבי שחרור של זיכרון מיותר)

תשובה:

```

MyString operator+(const MyString& a,const MyString& b) {
    char* as=a.getS();
    char* bs=b.getS();
    char* s=new char[strlen(as)+ strlen(bs)+1];
    for(int i=0;i<strlen(as);s[i]=as[i],i++);
    for(int i=0;i<strlen(bs);s[strlen(as)+i]=bs[i],i++);
    MyString t(s);
    Delete[] s;
    return t;
}

const MyString& MyString:: operator=(const MyString& a){
    delete[] s;
    s = new char[strlen(a.s)+1];
    strcpy(s,a.s);
    return *this;
}

MyString:: MyString(const MyString& a){
    s=null;
    *this=a;
}

```

שאלה 4 – Templates (20 נקודות)

```

template <class T>
class A {
public:

```

```

    A() {}
private:
    T i;
};
int main()
{
    A< A < A<int> > > > a;
    cout<<sizeof(a)<<endl;
}

```

- א. מהו הפלט של התוכנית לעיל, ומדוע? (10 נק')
- ב. מה יהיה הפלט של התוכנית לעיל אם נוסיף משתנים j, k מסוג T כחברי מחלקה של A (T i, j, k;)? מדוע? (10 נק')

תשובה:

- א. 4, הרי ה int גודלו 4 ולכן A<int> המכיל איבר יחיד בגודל 4 גודלו גם יהיה 4. וכך הלאה, A<A<int>> מכיל איבר יחיד מסוג A<int> שגודלו הוא 4 ולכן גודלו של A<A<int>> הוא 4. לא משנה כמה פעמים נמשיך ונעמיק את ההגדרה, תמיד יהיה מדובר באובייקט שמכיל איבר יחיד שגודלו 4 ולכן יהיה גודלו גם כן 4.
- ב. הפעם גודל האובייקט ב main יהיה 108, כי גודל האובייקט מסוג A<int> הוא 12 (מכיל 3 איברים מסוג int) ולכן גודל האובייקט A<A<int>> המכיל שלושה איברים בגודל 12 יהיה בגודל 36, ובהתאמה A<A<A<int>>> המכיל שלושה איברים בגודל 36 יהיה בגודל 108.

שאלה 5 (10 נק', כל סעיף 2 נק')

- א. מהו אלגוריתם גנרי על פי שיטת פולימורפיזם?
- ב. מהו אלגוריתם גנרי על פי שיטת templates?
- ג. מה משמעות ה const בפונקציה: const int& f(Point p) {...}?
- ד. אם יש לנו מתודות במחלקה שכל הזמן מקבלות את אותו הפרמטר, כנראה שכדאי שהפרמטר הזה יהיה....?
- ה. מה המילה השמורה friend מאפשרת לעשות?

תשובות:

- א. ניצור מחלקה אבסטרקטית עם מתודות pure virtual וכך נוכל לקבל כפרמטר בזמן ריצה כל אובייקט מסוג מחלקה שירשה את המחלקה האבסטרקטית שלנו ומימשה את כל המתודות חסרות המימוש. האלגוריתם הגנרי שלנו יבצע קריאות לאותן מתודות וכך אינו תלוי במימוש שלהם ועל כן הוא גנרי. דרך נוספת היא לכתוב את האלגוריתם במחלקה האבסטרקטית, וכל מחלקה שתירש את האבסטרקטית תצטרך לממש רק את המתודות הטהורות ואת האלגוריתם הגנרי קבלה בירושה.
- ב. האלגוריתם הגנרי יעשה שימוש ב template class T כלשהו שהתקבל כפרמטר.

- האלגוריתם יבצע קריאות למתודות הממושות ב T, וכך לא נהיה תלויים בטיפוס עליו אנו פועלים או באופן בו הוא מימש את המתודות שלו, ועל כן הוא גנרי.
- ג. ה int שחוזר הוא מוגן מפני כתיבה באותה שורת הקוד ש f חזרה בה. לדוגמא: לא ניתן לבצע $f(p)=5$ כי הרפרנס ל int מוגן כ const.
- ד. משתנה מחלקה
- ה. כאשר מצהירים על פונקציה גלובאלית כ friend בתוך מחלקה, לפונקציה זו יש גישה למשתנים הפרטיים של אותה המחלקה, וכאשר מצהירים על מחלקה X כ friend בתוך מחלקה Y, אז ל X יש גישה למשתנים הפרטיים של Y

שאלה 6: ירושה מרובה (10 נקודות)

האם קטע הקוד הבא מתקמפל, אם לא – מדוע?

אם כן – מה מודפס למסך?

```
class A{
public:
    virtual void m(){ cout<<"A::m()"<<endl;}
};

class B1:public virtual A{
public:
    virtual void m(){
        cout<<"B1::m()"<<endl;
    };
};

class B2:public virtual A{
public:
    virtual void m(){
        cout<<"B2::m()"<<endl;
    }
};

class D:public B1,B2{
};

int main(){
    A* d=new D();
    d->m();
}
```

תשובה: הקוד לא מתקמפל. כבר בנסיון לבנות את מחלקה D, בעת הנסיון לכתוב את הטבלה הוירטואלית שלה, לא ניתן לדעת מאיזו מחלקה לקחת את המימוש של המתודה הוירטואלית m, האם מ B1 או מ B2??