0600603099

שאלון + פתרון

המסלול האקדמי
המכללה למינהל

המסלול האקדמי המכללה למינהל

ביה"ס למדעי המחשב

ברקוד נבחן

ת.ז הסטודנט: _____

מספר חדר: _____

מספר נבחן: _____

מספר אסמכתא: _____

מבחן בקורס: תכנות מונחה עצמים

תאריך הבחינה: 15.08.13

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

משך הבחינה: 3.5 שעות

שם המרצה/ים:

אליהו חלסצ'י

שם המתרגל/ים:

חיים שפיר

שי צוויג

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות הכולל בבחינה: 6.

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{
   int x;
   public:
   A(){
      cout<<"A"<<endl;
   }
   ~A(){
      cout<<"~A"<<endl;
   }
};
class B:public A{
   A a;
   A* pa;
   B* pb;
   public:
   B(){
      cout<<"B"<<endl;
   }
   ~B(){
      cout<<"~B"<<endl;
   }
};
class C:public B{
   B b;
   C* pc;
   public:
   C(){
      cout<<"C"<<endl;
   }
   ~C(){
      cout<<"~C"<<endl;
   }
};
int main()
{
   C c;
   cout<<"---------------"<<endl;
   cout<<sizeof(float*)<<endl;
   cout<<sizeof(A)<<endl;
   cout<<sizeof(B)<<endl;
   cout<<sizeof(C)<<endl;
   cout<<"---------------"<<endl;
   return 0;
}
```

מהו הפלט של התוכנית? (10 נק')

תשובה:

```
A
A
B
A
A
B
C
---------------
4
4
16
36
---------------
~C
~B
~A
~A
~B
~A
~A
```

שאלה 2: ירושה ופולימורפיזם (30 נקודות)

אנו בונים משחק מחשב. נתונה המחלקה הבאה המגדירה דמות כללית במשחק, מחלקה זו מגדירה את כל המשותף לכל הדמויות השונות במשחק, כמו למשל המיקום שלהן במפה הנקבע ע"י x,y:

```
class GameCharacter{
    ...
  public:
  GameCharacter(int x,int y){ ... }
    ...
};
```

מחלקה זו ניתנה לנו בקוד סגור – היא אינה ניתנת לעריכה ולא ניתן להניח קיומן של מתודות שונות פרט למה שמופיע לעיל.

כמו כן, התחלנו לבנות את המחלקה הבאה המתארת חרב:

```
class Sword{
   int m_strength;
   char* m_name;
   public:
   Sword(const char* name, int strength){
      m_name=new char[strlen(name+1)];
      strcpy(m_name,name);
      m_strength=strength;
```

```
    }
    int getStrength() const {return m_strength;}
    ...
};
```

הערה: אין להוסיף בונה דיפולטיבי למחלקה Sword.

א. כיתבו copy Ctor עבור המחלקה Sword (3 נק')

ב. כיתבו (בצורה מונחית עצמים) את המחלקה Warrior (לוחם) ואת הבונה שלה, ע"פ ההגדרות הבאות: (10 נק')

a. לוחם הוא דמות משחק.

b. ללוחם יש חרב.

c. על הבונה לקבל פרמטרים שבאמצעותם יש לאתחל את הלוחם.

ג. כתבו פונקציית main המיצרת לוחם במיקום 5,6 בעל חרב בשם Excalibur עם עוצמה 10. (2 נק')

ד. נתון מערך של דמויות משחק **GameCharacter המאותחל בכל מיני סוגים שנוים של דמויות משחק. כתבו פונקציה (גלובאלית) המקבלת כפרמטרים את המערך, את גודלו של המערך, ופרמטר עוצמה. על הפונקציה להחזיר את מספר החרבות השייכות ללוחמים שעוצמתן גדולה מפרמטר העוצמה (15 נק')

לדוגמא אם פרמטר העוצמה היה 5, והלוחמים שבמערך שלנו מחזיקים חרבות בעוצמות הבאות: 4,5,5,10,10,8,3,2,1 אז הפונקציה שלנו צריכה להחזיר 3.

דגשים:

* פרט לפונקציה, ניתן להוסיף מתודות רק למחלקה Warrior
* הקפידו על תכנות מונחה עצמים נכון – לדוגמא, מתודה לא תחזיר ערך שלא שייך ישירות למחלקה שלה
* הקפידו על יעילות – כיצד פרמטרים מועברים
* שימו לב שלא כל הדמויות במערך הם לוחמים

פתרון:

א.

```
Sword(const Sord & a_sword){
    m_name=new char[strlen(a_sword.m_name+1)];
    strcpy(m_name,a_sword.m_name);
    m_strength=a_sword.m_strength;;
}
```

ב.

```
class Warrior : public GameCharacter{
   Sword sword;
   public:
   Warrior(int x,int y,const Sword& s) : GameCharacter(x,y) , sword(s) { }
};
```

ג.

```
int main() {
   Sword s("Excalibur",10);
   Warrior w(5,6,s);
    return 0;
}
```

ד.

למחלקה Warrior נוסיף את המתודה הבאה:

```
const Sword& Warrior::getSword(){ return sword;}
```

ונכתוב את הפונקציה:

```
int count_strengths(GameCharacter** arr, int size, int strength){
    int count=0;
    for(int i=0;i<size;i++){
            if(typeid(arr[i]) == typeid(Warrior*)) {        // if it is a warrior
                    Warrior* w = (Warrior*)arr[i];          // convert to warrior
                    if( (w->getSword()).getStrength() > strength)
                            count++
            }
    }
    return count;
}
```

שאלה 3: Operator Overloading – Const (20 נקודות)

נתונה המחלקה הבאה המיצגת אדם

```
class Person{
   char* name;
   public:
   Person(const char* a_name){
        name = new char[strlen(a_name)+1];
        strcpy(name,a_name);
   }
   const char* getName() const {return name;}
};

int main(){
 Person alice("Alice"),bob("Bob");
 Person charlie("Charlie");
 charlie = alice + bob;
 cout<< charlie.getName()<<endl; // Alice Bob
}
```

ממשו את המתודות והאופרטורים הדרושים לכך שהדוגמא ב main תעבוד, כאשר את אופרטור השרשור (+) יש לממש **כפונקציה גלובאלית** ולא כמתודה.

דגשים:

* שימו לב לגבי שחרור של זיכרון מיותר
* שימו לב לכתיבת קוד נכון – יש מקרים שה main לא בודק, ובכל זאת אתם נדרשים לחשוב עליהם כאשר ממשים אופרטורים בצורה נכונה. (אופן קבלה של פרמטרים, והחזרה של ערכים)
* שימו לב שבין המילים המשורשרות יש רווח

תשובה:

```
Person operator+(const Person & a,const Person & b) {
    char* as=a.getName();
    char* bs=b.getName ();
    char* s=new char[strlen(as)+ strlen(bs)+2];
    for(int i=0;i<strlen(as);s[i]=as[i],i++);
    s[strlen(a)]=' ';
    for(int i=0;i<strlen(bs);s[strlen(as)+1+i]=bs[i],i++);
    Person t(s);
    Delete[] s;
```

```
    return t;
}

const Person& Person:: operator=(const Person& a){
    if(this!=&a){
            delete[] name;
            name = new char[strlen(a.name)+1];
            strcpy(name,a.name);
    }
    return *this;
}
Person:: Person (const Person & a){
    name=null;
    *this=a;
}
```

שאלה 4 – Templetes (20 נקודות)

נתונה פונקציית ה main הבאה:

```
int main() {
   int i=2;
   float f=2.2;
   SQR sqr;              // notice!!!
   sqr(i);
   sqr(f);
   cout << i << endl;  // 4
   cout << f << endl;  // 4.84
   return 0;
}
```

למרות ש i ו f הינם טיפוסים שונים, הפעולה sqr מעלה את ערכם בריבוע...

א. מהו sqr? יש לדייק ולפרט (5 נק')

ב. מהו SQR? יש לדייק ולפרט (5 נק')

ג. ממשו את הנדרש כדי שהקוד ב main יעבוד. (10 נק')

תשובה: