

מטלת מנחה (ממ"ן) 11

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה: יחידה 2 – שפת C++

משקל המטלה: 3

מספר השאלות: 3

מועד אחרון להגשה: 24.11.2022

סמסטר: 2023א

שימו לב:

את המטלה יש להגיש באמצעות מערכת המטלות המקוונת באתר הבית של הקורס בלבד
את התשובה יש להגיש בקבצים בהתאם למפורט בשאלות.

שאלה 1 (10%)

מה יודפס בהרצת הקוד הבא ולמה? הסבירו היטב את תשובתכם והתייחסו למנגנון הפולימורפיזם כפי שהוא ממומש בשפת C++.

```
#include <iostream>

class Foo
{
public:
    Foo() { baz(); }
    virtual void baz() { std::cout << "Foo::baz()" << std::endl; }
};

class Bar : public Foo
{
public:
    Bar() {}
    virtual void baz() { std::cout << "Bar::baz()" << std::endl; }
};

int main()
{
    Foo *pFoo = new Bar();
    delete pFoo;
    return 0;
}
```

את הפתרון יש להגיש בקובץ Word או PDF.

שאלה 2 (10%)

לפניכם המחלקה Point וקוד העושה בה שימוש. הקובץ point.cpp באתר הקורס, הריצו אותו. האם התקבלה התוצאה לה ציפיתם? מצאו את הבאג, תקנו אותו והסבירו את התיקון שלכם.

```
#include <iostream>

class Point
{
    int* _coord;
public:
    Point()
    {
        _coord = new int[2];
        _coord[0] = _coord[1] = 0;
    }

    Point(int x, int y)
    {
        _coord = new int[2];
        _coord[0] = x;
        _coord[1] = y;
    }

    Point(const Point& other)
    {
        _coord = other._coord;
    }

    ~Point()
    {
        delete _coord;
    }

    void setX(int value) { _coord[0] = value; }
    void setY(int value) { _coord[1] = value; }

    friend std::ostream& operator<<(std::ostream& os, const Point& p)
    {
        os << "(" << p._coord[0] << "," << p._coord[1] << ")";
        return os;
    }
};

int main()
{
    Point p1(1, 2);
    Point p2 = p1;

    p2.setX(5);

    std::cout << "p1=" << p1 << std::endl;
    std::cout << "p2=" << p2 << std::endl;

    return 0;
}
```

}

הגשה : קובץ עם הקוד המתוקן ומסמך pdf או word. רצוי לצרף אותם לקובץ zip.

שאלה 3 (80%)

בתרגיל זה נממש רשת חברתית בשם USocial

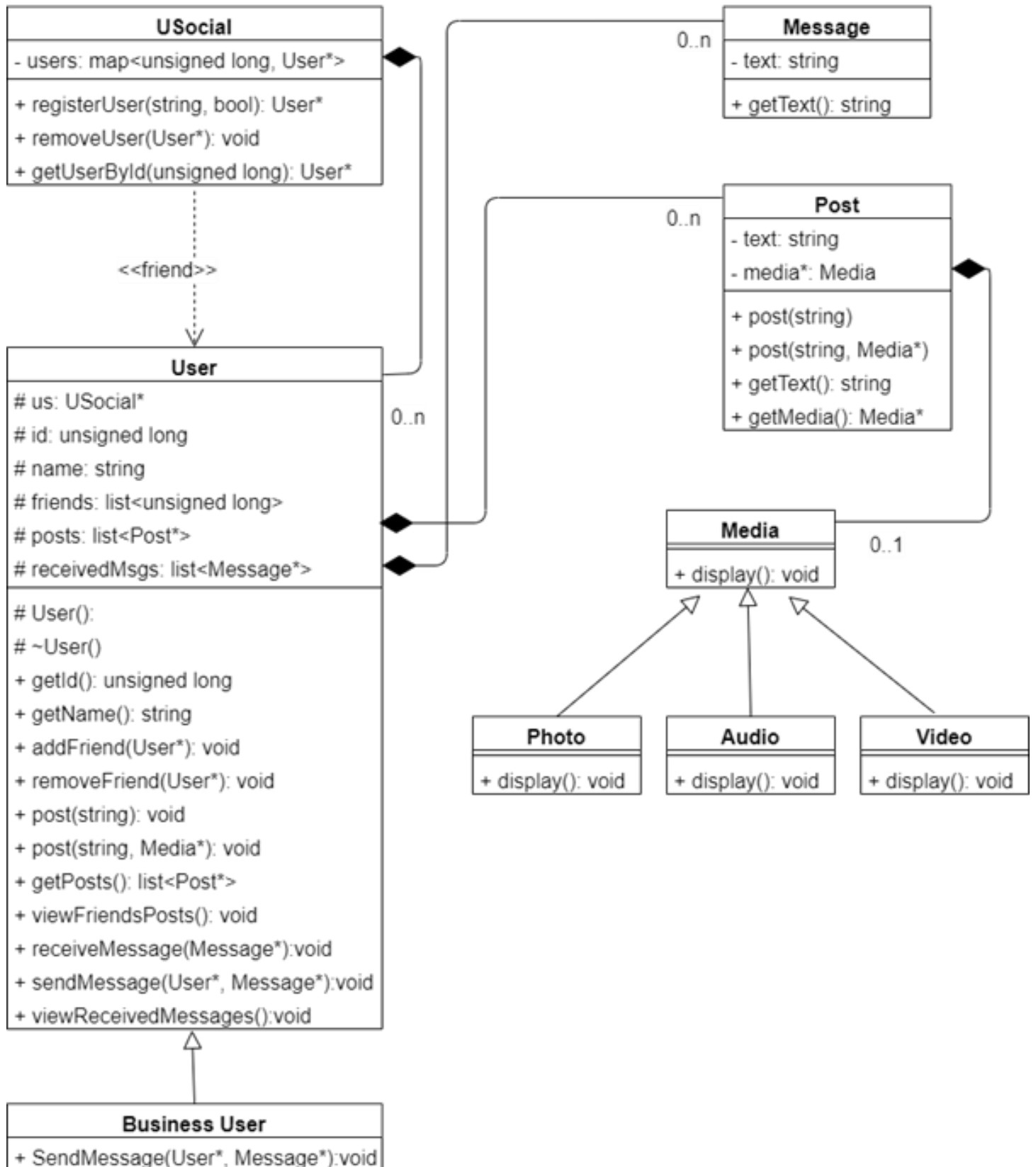
הרשת החברתית תכלול את המחלקות הבאות

שם	תיאור
US	מחלקה המתארת רשת חברתית
Media	מחלקה אבסטרקטית טהורה המייצגת אובייקט מדיה
Photo	מחלקה יורשת מ-Media ומייצגת תמונה
Audio	מחלקה יורשת מ-Media ומייצגת קול (קובץ אודיו)
Video	מחלקה יורשת מ-Media ומייצגת וידאו
Post	מחלקה המייצגת פוסט (רשומה חדשה). פוסט חייב לכלול טקסט ויכול לכלול אובייקט מדיה אחד. חשוב! פוסטים שייכים למשתמש שפרסם אותם, אבל חבריו יכולים לבקש לראות אותם. כלומר, אם אובייקט A פרסם פוסט, הוא יהיה שייך ל-A ובאחריות A לשחרר את הזיכרון בתום ריצת התוכנית.
User	מחלקה המייצגת משתמש רגיל.
BusinessUser	מחלקה המייצגת משתמש עסקי
Message	מחלקה המייצגת הודעה. הודעה חייבת לכלול טקסט - משתמש רגיל יכול לשלוח הודעה רק למשתמשים ברשימת החברים שלו. - משתמש עסקי יכול לשלוח הודעה לכל משתמש אחר. חשוב! הודעות שייכות לאובייקט אליו הן נשלחו – כלומר, אם אובייקט A שולח הודעה לאובייקט B, ההודעה תישמר בזיכרון של B והוא גם אחראי לניקוי הזיכרון.

לפניכם דיאגרמת UML¹ המתארת את העיצוב הנדרש (Design) :

¹ דיאגרמת UML

היא שפת מפרט תקנית לעיצוב מונחה-עצמים



דגשים:

- א. ממשו את המחלקות השונות לפי דיאגרמת ה-UML. יש לממש את כל הפונקציות אך מותר להוסיף פונקציות כרצונכם.
- ב. אין צורך לכתוב מידע לדיסק או להשתמש בבסיסי נתונים, כל האובייקטים ייווצרו בזמן ריצה בזיכרון RAM.
- ג. שימו לב להקצאות זיכרון דינמיות ושיחרור הזיכרון בצורה נכונה. **מרבית האובייקטים מוגדרים כמצביעים.**
- ד. חשבו על **פולימורפיזם** – אילו פונקציות צריכות להיות וירטואליות?
- ה. מרבית הפונקציות מוגדרות void אולם עשויות להכשל. **עשו שימוש בחריגות.**
- ו. מומלץ (אבל לא חובה) לעשות שימוש בספריות STL, שימו לב במיוחד לספריה `<algorithm>`.
- ז. ניתן ורצוי להשתמש ביכולות C++11 (לדוגמה פונקציות מסוג למדה, שימוש ב- `auto` וכ'...).
- ח. שימו לב להרשאות גישה של המשתנים והפונקציות, הסימון המקובל ב-UML הוא:

+	ציבורי
-	פרטי
#	מוגן (protected)

- ט. לפי הרשאות הגישה של הבנאי והמפרק של המחלקה User הדרך היחידה לייצר אובייקטים כאלה היא באמצעות המחלקה USocial. מדוע?
- י. מחלקת User ו-USocial צריכות להכיר אחת את השניה (הן גם מחלקות חברות). **שימו לב להפניות מעגליות!**
- יא. בפונקציה `viewFriendsPosts` : User יש להדפיס את הפוסטים של המשתמשים ברשימת החברים. אם פוסט מכיל מדיה, יש להדפיס גם אותה.
- יב. בפונקציה `viewReceivedMessages` : User יש להדפיס את ההודעות של **אותו משתמש** (ההודעות שנשלחו אליו).
- יג. במחלקות היורשות מ-Media יש לממש את הפונקציה `display` ולהדפיס, "image", "audio", "video" בהתאמה.
- יד. הקפידו על תיעוד של הפונקציות (comments).
- טו. **מקרי קצה** חשוב לתת דגש למקרי קצה, לדוגמה: לא ניתן להוסיף משתמש כחבר שכבר מופיע ברשימת החברים שלנו, במצב כזה תיזרק חריגה.

הגשה:

1. עליכם להגיש רק את קבצי הקוד (כלומר קבצי h ו-cpp). מצורפים יחד לקובץ zip. **שימו לב!** על התוכנית להתקמפל ולרוץ בצורה תקינה (ללא צורך בתוספות קבצים, ללא קריסות)
2. יש לכלול גם קובץ ראשי (שמכיל את פונקציית main) ושכולל ריצות לדוגמה.
3. עבודתכם תיבדק במערכת הפעלה חלונות, באמצעות Visual Studio ולכן מומלץ לעבוד עם סביבה זו או עם סביבת Eclipse.
4. תוכלו להעזר בקוד הבא לבדיקות בסיסיות של הפיתרון שלכם.

```
#include "USocial.h"
#include "User.h"

#include <exception>
#include <iostream>
```

```

int main()
{
    USocial us;

    User* u1 = us.registerUser("Liron");
    User* u2 = us.registerUser("Yahav");
    User* u3 = us.registerUser("Shachaf");
    User* u4 = us.registerUser("Tsur", true);
    User* u5 = us.registerUser("Elit");

    u1->post("Hello world!");
    u2->post("I'm having a great time here :)", new Audio());
    u3->post("This is awesome!", new Photo());

    u5->addFriend(u1);
    u5->addFriend(u2);

    u5->viewFriendsPosts();    // should see only u1, u2 s' posts

    u4->sendMessage(u5, new Message("Buy Falafel!"));
    u5->viewReceivedMessages();

    try
    {
        u3->sendMessage(u5, new Message("All your base are belong to
us"));
    }
    catch (const std::exception& e)
    {
        std::cout << "error: " << e.what() << std::endl;
    }
    u5->viewReceivedMessages();

    u3->addFriend(u5);
    u3->sendMessage(u5, new Message("All your base are belong to us"));
    u5->viewReceivedMessages();

    return 0;
}

```

בהצלחה!