ה א ו נ י ב ר ס י ט ה ה פ ת ו ח ה

20937

תכנות מערכות דפנסיבי

חוברת הקורס –חורף 2021א

כתב: רועי מימרן

ספטמבר 2020 – סמסטר חורף – תשפ"א

פנימי – לא להפצה.

תוכן העניינים

אל הסטודנט א

1. לוח זמנים ופעילויות ג
2. תיאור המטלות ה
מבנה המטלות ה
ניקוד המטלות ה

3. התנאים לקבלת נקודות זכות בקורס ו

ממ״ח 01 1
ממ״ן 11 3
ממ״ן 12 5
ממ״ן 13 7
ממ״ן 14 9

אל הסטודנט,

אני מקדם את פניך בברכה עם הצטרפותך אל הלומדים בקורס "תכנות מערכות דפנסיבי", בסמסטר הבכורה של קורס זה.
בחוברת זו תמצא את הדרישות לקבלת נקודות זכות בקורס, לוח הזמנים ומטלות.

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, אותם מפרסם/מת מרכז/ת ההוראה.
בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס.
פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ם בכתובת:
http://telem.openu.ac.il
מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספריה באינטרנט www.openu.ac.il/Library.

שעות הייעוץ שלי בכל יום ב', בשעות 11:00-12:30, בטלפון 09-7781270. פגישה רצוי לתאם מראש. ניתן לפנות גם בדואר אלקטרוני: roymim@openu.ac.il

מילת התנצלות לסטודנטיות בקורס: פניות המופיעות בחומר הלימוד מנוסחות בלשון זכר - זהו למרבה הצער הנוהג המקובל. הפניות האלו מכוונות, כמובן, לכל קוראי החומר.

לתשומת לב הסטודנטים הלומדים בחו"ל:

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.
הפרטים החיוניים על הקורס נכללים בחוברת הקורס וכן באתר הקורס.
מומלץ מאד להשתמש באתר הקורס ובכל אמצעי העזר שבו וכמובן לפנות אלינו במידת הצורך.

אני מאחל לך לימוד פורה ומהנה.

בברכה,

רועי מימרן
מרכז ההוראה בקורס

1. לוח זמנים ופעילויות (מס' קורס 20937/א2021)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	23.10.2020-18.10.2020	יחידה 1	מפגש 1	
2	30.10.2020-25.10.2020	יחידה 2		
3	06.11.2020-01.11.2020	יחידה 2	מפגש 2	
4	13.11.2020-08.11.2020	יחידה 2		ממ"ח 01 8.11.2020
5	20.11.2020-15.11.2020	יחידה 3	מפגש 3	ממ"ן 11 15.11.2020
6	27.11.2020-22.11.2020	יחידה 3		
7	04.12.2020-29.11.2020	יחידה 4	מפגש 4	ממ"ן 12 29.11.2020
8	11.12.2020-06.12.2020 (ו חנוכה)	יחידה 4	מפגש 5	

** התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".*

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	18.12.2020-13.12.2020 (א-ו חנוכה)	יחידה 5		
10	25.12.2020-20.12.2020	יחידה 5	מפגש 6	ממ"ן 13 20.12.2020
11	01.01.2021-27.12.2020	יחידה 6		
12	08.01.2021-03.01.2021	יחידה 6	מפגש 7	ממ"ן 14 3.1.2021
13	15.01.2021-10.01.2021	יחידה 7		
14	22.01.2021-17.01.2021	סיכום	מפגש 8	ממ"ן 15 **חובה** 31.1.2021

מועדי בחינות הגמר יפורסמו בנפרד

** התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".*

** התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".*

2. תיאור המטלות

קרא היטב עמודים אלו לפני שתתחיל לענות על השאלות

פתרון המטלות הוא חלק בלתי נפרד מלימוד הקורס - הבנה מעמיקה של חומר הלימוד דורשת תרגול רב. המטלות תיבדקנה על-ידי המנחה ותוחזרנה לך בצירוף הערות המתייחסות לתשובות.

מבנה המטלות

כל מטלה מורכבת מכמה שאלות. בראש כל שאלה מצוין משקלה היחסי בקביעת ציון המטלה.

את הפתרונות לממ"ח 01 עליך להגיש באמצעות סימון התשובה הנכונה ביותר מבין האפשרויות במערכת שאילתא.
את הפתרונות לממ"נים עליך להגיש בקובץ שיוגש באמצעות מערכת המטלות. בחלק מהמטלות תהיינה הנחיות ספציפיות לגבי אופן ההגשה.

אם השאלה בממ"ן אינה ברורה לך, אל תהסס להתקשר אל אחד מהמנחים (בשעות הייעוץ הטלפוני שלו) לצורך קבלת הסבר.

המטלות מלוות את יחידות הלימוד בקורס. להלן פירוט המטלות והיחידות שאליהן מתייחסת כל מטלה. בחלק מהמטלות תופענה גם שאלות המתייחסות ליחידות קודמות, שכבר נלמדו.

ניקוד המטלות

כל מטלה נקבע משקל; ניתן לצבור עד 30 נקודות. חובה להגיש מטלות במשקל של 20 נקודות לפחות.

ללא צבירת 8 נקודות בהגשת מטלות לא ניתן יהיה לגשת לבחינת הגמר

להלן פירוט הניקוד לכל מטלה:

ממ"ן	ניקוד
01	2
11	3
12	4
13	3
14	4
15	14

3. התנאים לקבלת נקודות זכות בקורס

א. הגשת מטלת החובה בקורס – ממ"ן 15 בציון 50 לפחות.

ב. ציון של לפחות 60 נקודות בבחינת הגמר.

ג. ציון סופי משוקלל בקורס של 60 נקודות לפחות.

לתשומת לבכם!

פתרון המטלות הוא מרכיב מרכזי בתהליך הלמידה, לכן מומלץ שתשתדלו להגיש מטלות רבות ככל האפשר, כולל מטלות שעליהן תצליחו להשיב באופן חלקי בלבד.

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:

בחישוב הציון הסופי נשקלל את כל המטלות שציוניהן גבוהים מהציון בבחינת הגמר. ציוני מטלות כאלה תורמים לשיפור הציון הסופי.
ליתר המטלות נתייחס במידת הצורך בלבד. מתוכן נבחר רק את הטובות ביותר עד להשלמת המינימום ההכרחי לעמידה בתנאי הגשת מטלות. משאר המטלות נתעלם.

זכרו! ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.

מטלת מחשב (ממ״ח) 01

הקורס: 20937 – תכנות מערכות דפנסיבי

חומר הלימוד למטלה: יחידה 1 – מבוא לתכנות דפנסיבי

מספר השאלות: 10 **משקל המטלה:** 2

סמסטר: 2021א **מועד אחרון להגשה:** 8.11.2020

את התשובות לממ״ח יש לשלוח באמצעות מערכת **שאילתא** בכתובת www.openu.ac.il/sheilta

בממ״ח זה עליכם לבחור את התשובה הנכונה ביותר על-בסיס החומר שנלמד ביחידה.

1. ע״פ פרק 1 בספר, במה מתמקדות הבדיקות שעשו עד לאחרונה רוב יצרני התוכנה?
   א. אבטחה, מניעת פרצות ומניעת השתלטות עוינת על המחשב.
   ב. נושאים שיווקיים: תכונות, זמינות ויציבות כללית של התוכנה.
   ג. עיצוב גרפי וחווית משתמש של התוכנה.
   ד. מניעת האגים ונפילות של התוכנה.
2. א. כיצד נקראות בדיקות אשר מבצעות רק מניפולציה על הקלט והממשקים החיצוניים של התכנית:
   א. מבחן התוצאה.
   ב. בדיקה חיצונית.
   ג. מבחני קופסה שחורה.
   ד. ביקורת תוכנה.
3. מה המשמעות של הגנה לעומק (Defense in Depth)?
   א. מנגנון הגנה שאין לו פרצות.
   ב. אבטחה במספר שכבות; כל פשרה בהגנה בנקודה אחת תכוסה בשכבה אחרת.
   ג. הגנה באמצעות אזור חיצוני (DMZ).
   ד. הגנה באמצעות חסימת גישה לרשת.
4. מה הקשר המומלץ בין מודולים ומחלקות שונים, מול מרכיבים שונים בתוך אותו מודול או מחלקה?
   א. צמידות חזקה בין מחלקות שונות, לכידות חזקה בתוך אותה מחלקה.
   ב. צמידות חלשה בין מחלקות שונות, לכידות חזקה בתוך אותה מחלקה.
   ג. צמידות חזקה בין מחלקות שונות, לכידות חלשה בתוך אותה מחלקה.

ד. צמידות חלשה בין מחלקות שונות, לכידות חזקה בתוך אותה מחלקה.

שאלות 5-8 מתייחסות לדוגמא הבאה. נניח שהנכם מתכננים אתר ואפליקציית היכרויות. לכל משתמש יש אפשרות לפתוח כרטיס. בכרטיס ישנם פרטים גלויים על המשתמש: גיל, מידע שרשם על ההשכלה והעבודה שלו ובכלל על עצמו, תמונה אחת או יותר (עשויה להיות גלויה או לא ע"פ בחירת המשתמש), וכמו כן יש פרטים חסויים: כתובת דוא"ל, אמצעי תשלום, העדפות חיפוש ועוד. קיים מערך קשרים בין משתמשים, המתאפשר לאחר ששני המשתמשים ביצעו אישור הדדי (הזזה ימינה באפליקציה או סימון V), ובמקרה כזה מתאפשרת פתיחת צ'ט פרטי בין המשתמשים, שליחת פרטים, צפייה חוזרת בכרטיס מתוך רשימת קשרים (ולא רק באופן אקראי).

5. ע"פ פרק 2 בספר, מהי הדרך המומלצת להגדיר את פרטי אמצעי התשלום בתוכנה בצד השרת:

א. שדה גלוי במחלקה שמציינת את המשתמש.

ב. שדה פרטי במחלקה שמציינת את המשתמש.

ג. מחלקה נפרדת לפרטי אמצעי תשלום.

ד. קובץ נפרד וחסוי במחשב לפרטי אמצעי התשלום.

6. מהי הדרך המומלצת להגדיר קשר בין משתמשים:

א. לכידות חזקה בין שתי מחלקות של משתמשים.

ב. מחלקה נפרדת לקשר בין המשתמשים, הכוללת קשר עם מחלקות המשתמשים.

ג. שדה נוסף במחלקה של המשתמש הכולל קשרים למשתמשים אחרים.

ד. שיטה מיוחדת של המשתמשים ליצירת קשר ביניהם.

7. קיימים במערכת משתמשי פרימיום (אשר משלמים תשלום נוסף וזכאים לשירותים נוספים). כיצד כדאי לממש אותם:

א. אלו משתמשים רגילים עם שדות בוליאניים שדלוקים עבורם.

ב. באמצעות מחלקת ירושה הכוללת שדות ושיטות נוספות.

ג. באמצעות רקע מיוחד שיופיע בתצוגת הכרטיסים האישיים שלהם, על מנת להבליט.

ד. הבלטה באמצעות הופעה ראשונים בתוצאות חיפוש.

8. מדוע לדעתך המערכת עשויה לחסום משתמשים המסמנים "ימינה" ליותר מדי משתמשים אחרים בשעה אחת:

א. חשש להצפת המערכת בתנועת יתר כדי לחסום אותה, או רצון לנצל את הצ'טים כדי לכתוב פרסומות.

ב. המערכת מאמינה בקשרים מונוגמיים ונאמנות, זה לא יעבוד אם ממקבלים יותר מדי קשרים.

ג. חשש לסתימת הזכרון באפליקציה לטלפונים ניידים.

ד. חשש לסתימת הזכרון בשרת בגלל התמלאות ערוצי הקשרים והצ'טים.

9. מה היתרונות של פרוטוקולי הצפנה מסוג RSA, SSL?

א. הצפנה המורכבת והקשה ביותר לפריצה שקיימת בשימוש כיום.

ב. עמידים לפריצה באמצעות מחשב-על עם מעבד וקטורי.

ג. עמידים לפריצה בעתיד גם ע״י מחשב קוואנטי.

ד. פרוטוקולים בדוקים ואמינים, מערכים שהסיכוי לבעיה בפרוטוקול שלהם נמוך מאד.

10. התקפות מניעת שירות (Denial of Service) הן :

א. מצב שבו השרת נפל ואינו מגיב כלל.

ב. מצב שבו המשתמש נחסם ואינו מצליח לתקשר לשרת.

ג. מצב בו השרת מוצף באופן מלאכותי בהמון פניות בו זמנית כך שאינו מצליח להגיב לכולן.

ד. מצב שבו תוכנת השרת מופעלת בארגז חול (Sandbox).

מטלת מנחה (ממ"ן) 11

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה: יחידה 2 – שפת C++

מספר השאלות: 3 **משקל המטלה:** 3

סמסטר: 2021א **מועד אחרון להגשה:** 15.11.2020

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.

שאלה 3 (80%)

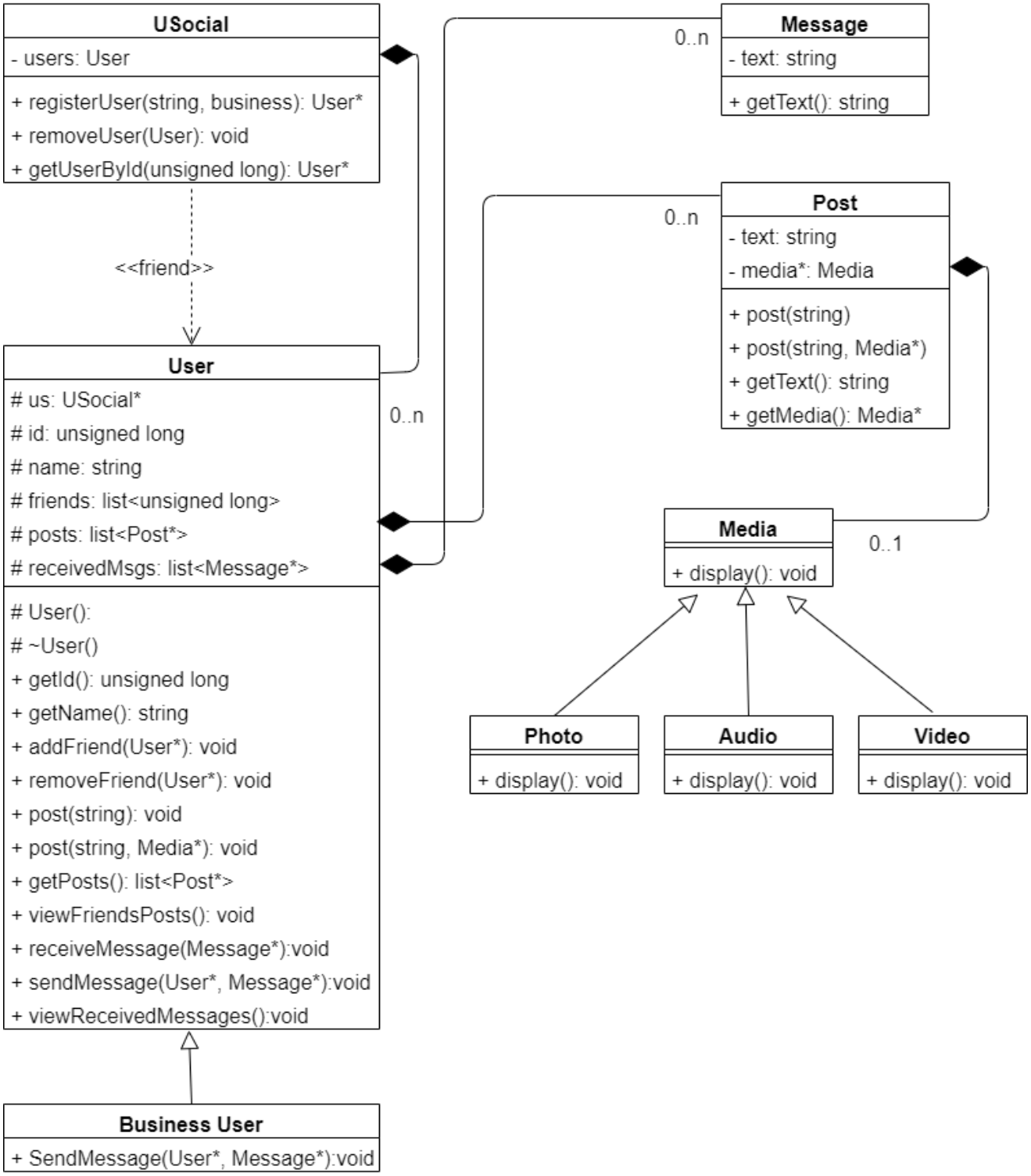
בתרגיל זה נממש רשת חברתית בשם USocial

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 פוֹסְט (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 UML[1] המתארת את העיצוב הנדרש (Design):

[1] דיאגרמת UML היא שפת מפרט תקנית לעיצוב מונחה-עצמים


USocial
- users: User
+ registerUser(string, business): User*
+ removeUser(User): void
+ getUserById(unsigned long): User*
<<friend>>
User
us: USocial*
id: unsigned long
name: string
friends: list<unsigned long>
posts: list<Post*>
receivedMsgs: list<Message*>
User():
~User()
+ getId(): unsigned long
+ getName(): string
+ addFriend(User*): void
+ removeFriend(User*): void
+ post(string): void
+ post(string, Media*): void
+ getPosts(): list<Post*>
+ viewFriendsPosts(): void
+ receiveMessage(Message*):void
+ sendMessage(User*, Message*):void
+ viewReceivedMessages():void
Business User
+ SendMessage(User*, Message*):void
0..n
0..n
Message
- text: string
+ getText(): string
0..n
Post
- text: string
- media*: Media
+ post(string)
+ post(string, Media*)
+ getText(): string
+ getMedia(): Media*
0..1
Media
+ display(): void
Photo
+ display(): void
Audio
+ display(): void
Video
+ display(): void

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 יווצרו 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 (לדוגמא פונקציות מסוג למדה, שימוש ב- auto וכו'..).
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 User::viewFriendsPosts יש להדפיס את הפוסטים של המשתמשים ברשימת החברים. אם פוסט מכיל מדיה, יש להדפיס גם אותה.
יב. בפונקציה User::viewReceivedMessages יש להדפיס את ההודעות של **אותו משתמש** (ההודעות שנשלחו אליו)
יג. במחלקות היורשות מ- Media יש לממש את הפונקציה display ולהדפיס "image", "video", "audio" בהתאמה.
יד. הקפידו על תיעוד של הפונקציות (comments).
טו. **מקרי קצה**
חשוב לתת דגש למקרי קצה, לדוגמא: 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.).
שימו לב! על התוכנית להתקמפל ולרוץ בצורה תקינה (ללא צורך בתוספות קבצים, ללא קריסות)
2. יש לכלול גם קובץ ראשי (שמכיל את פונקצית main) ושכולל ריצות לדוגמא.
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!

מטלת מנחה (ממ"ן) 12

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה: יחידה 3 – חולשות אבטחה בשפת C++

מספר השאלות: 2 **משקל המטלה:** 4

סמסטר: 2021א **מועד אחרון להגשה:** 29.11.2020

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 (20%)

בחברת האשראי "קשה", לקוח יכול להיות בעל קרדיט הנע בין 100- ל- 1000 שקלים.

לקראת החגים, החליטו בחברה לצאת במבצע קידום מכירות ולשלוח מתנה ללקוחות בעלי קרדיט הגדול מ- 750 שקלים.

להלן הקוד לבדיקה האם לקוח זכאי למתנה:

```
bool is_entitled_for_promotional_gift(int ID)
{
	unsigned int bound = 750;
	int credit = get_credit(ID);
	return (credit >= bound);
}
```

ליעל הסטודנטית, קרדיט מאוד נמוך בחברת האשראי. מה עליה לעשות כדי שתוכל לזכות במתנה המיוחלת?

א. מצאו את החולשה, הגדירו אותה והציעו דרך לתקוף את המערכת
ב. תקנו את הקוד כך שההתקפה לא תעבוד
ג. כתבו מסמך המתאר את החולשה, ההתקפה והתיקון.

הגשה: מסמך בפורמט pdf או word

שאלה 2 (80%)

לפניכם תוכנה המדפיסה ל- stdout את הקלט שלה. הקוד זמין גם בקובץ mmn02-q2.cpp באתר הקורס.

1. קמפלו את הקוד, הריצו אותו והבינו כיצד הוא עובד.
2. מצאו חולשה והשתמשו בה על מנת לקרוא לפונקציה unreachable.
3. תקנו את הקוד כך שההתקפה לא תעבוד.
4. כתבו מסמך מחקר עם הסבר על החולשה, ההתקפה וההגנה.

```
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <string>

////////////////////////////////////////////////////////////////////////////////
/////////
//
// -- IMPORTANT! --
//
// for this exercise to run correctly do the following:
//
// a. Disable ASLR:
//     VS:    Configuration  Properties->Linker->Advanced  ->  "Randomized  Base
Address"
//     g++: disabled by default in gdb
//
// b. Set the target binary to x86
//     VS: Build -> Configuration Manager -> Active solution platform -> X86
//     g++: -m32 flag    (if fails try: sudo apt-get install gcc-multilib g++-
multilib)
//
// c. Debug mode:
//     VS: Build -> Configuration Manager -> Active solution configuration ->
Debug
//     g++: -g3 flag   (maximal debug information)
//
////////////////////////////////////////////////////////////////////////////////
/////////


#define PROGRAM_NAME "echoutil"
#define VERSION "1.0"

#define VERY_SECRET_PASSWORD "Cowabunga!"

class Handler
{
   virtual void unreachable()
   {
      printf("%s", VERY_SECRET_PASSWORD);
      exit(0);
   }

   virtual void helper(const char *str)
   {
      std::string s = "0" + std::string(str);
      unsigned int x = std::stoul(s, nullptr, 16);
      printf("%c", x);
   }

public:
```

```
   void interpret(const char* str)
   {
      helper(str);
   }
};


void usage(int status)
{
   fputs("Echo the STRING(s) to standard output\n"
      "\n"
      "\t-n   do not output the trailing newline\n"
      "\t-e   enable interpretation of backslash escapes\n"
      "\n"
      "\tIf - e is in effect, the following sequences are recognized : \n"
      "\t\t\\xHH     byte with hexadecimal value HH(1 to 2 digits)\n"
      , stdout);

   exit(status);
}

void handle_escape(const char* str)
{
   struct
   {
      char buffer[16] = { 0 };
      Handler h;
   } l;

   // copy only the characters after the escape char
   const char* s = str;
   char* p = l.buffer;
   s++;
   while (*s)
      *p++ = *s++;

   // handle different options
   switch (l.buffer[0])
   {
   case 'x':
      l.h.interpret(l.buffer);
      break;

   default:
      fputs(str, stdout);
   }
}

char* dupenv(const char* varname)
{
#if defined(_WIN32)

   char* buff = NULL;
```

```
   size_t cnt;
   if (_dupenv_s(&buff, &cnt, varname) != 0)
      return NULL;
   return buff;

#elif defined(__linux__)

   const char* s = getenv(varname);
   if (!s)
      return NULL;
   return strdup(s);

#endif
}
int main(int argc, char** argv)
{
   bool display_return = true;
   bool do_escape = false;

   char* env = dupenv("ECHOUTIL_OPT_ON");
   bool allow_options = env != NULL;
   free(env);

   if (allow_options && argc == 2)
   {
      if (strcmp(argv[1], "--help") == 0)
         usage(EXIT_SUCCESS);

      if (strcmp(argv[1], "--version") == 0)
      {
         fprintf(stdout, "%s version %s\n", PROGRAM_NAME, VERSION);
         exit(EXIT_SUCCESS);
      }
   }

   --argc;
   ++argv;

   if (allow_options)
   {
      while (argc > 0 && *argv[0] == '-')
      {
         const char* temp = argv[0] + 1;
         size_t i;
         for (i = 0; temp[i]; i++)
            switch (temp[i])
            {
            case 'e': case 'n':
               break;
            default:
               goto just_echo;
            }
         if (i == 0)
            goto just_echo;
```

```
         // options are valid
         while (*temp)
            switch (*temp++)
            {
            case 'e':
               do_escape = true;
               break;

            case 'n':
               display_return = false;
               break;
            }

         argc--;
         argv++;
      }
   }

just_echo:

   while (argc > 0)
   {
      const char* s = argv[0];

      if(do_escape && s[0] == '\\')
         handle_escape(s);
      else
         fputs(argv[0], stdout);

      argc--;
      argv++;
      if (argc > 0)
         putchar(' ');
   }

   if (display_return)
      putchar('\n');

   exit(EXIT_SUCCESS);
}
```

דגשים:

א. עבור תרגיל זה יש **לבטל את מנגנון ה- ASLR** ולבנות את הקוד ב- **32 סיביות (x86)**
ב. קמפלו את הקוד בקונפיגורצית **debug** ועשו שימוש בדבאגר (מספיק שההתקפה תעבוד עם דבאגר).
ג. עבודתכם תיבדק במ״ה לינוקס (Ubuntu), באמצעות gcc ולכן מומלץ לעבוד עם סביבה זו.

הגשה: קובץ עם הקוד המתוקן ומסמך מחקר בפורמט pdf או word.

בהצלחה!

מטלת מנחה (ממ"ן) 13

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה: יחידה 4 – שפת פייתון ומטא-תכנות

מספר השאלות: 3 **משקל המטלה:** 3

סמסטר: 2021א **מועד אחרון להגשה:** 20.12.2020

שאלה 1 (20%)

ענו על כל שאלה באמצעות קובץ תכנית בן עד 10 שורות, והגישו את הקובץ למערכת המטלות.
כמובן יש להימנע מהצבת ערכים מפורשים בפתרון ויש לפתור באמצעות עיבוד של נתוני הפתיחה.

א. לכל פועל (מילה) ברשימה words, יש להעביר לזמן עבר באנגלית באופן הבא: להוסיף d בסוף המילה אם היא מסתיימת באות e, או להוסיף את האותיות ed אחרת. שמרו את המילים בצורות העבר ברשימה בשם past_tense. את הפתרון יש להגיש בקובץ past.py:

```
words = ["adopt", "bake", "beam", "cook", "time", "grill", "waved", "hire"]

# put your code below
```

ב. במחרוזת הבאה מופיעים מספר מ"מ גשם שירדו בחודשים שונים. חודש שירדו בו מעל 75 מ"מ נחשב גשום. ספרו את מספר החודשים הגשומים ברשימה ושמרו את התוצאה במשתנה num_rainy_months. את הפתרון יש להגיש בקובץ rain.py:

```
Rainfull_mi = "45, 65, 70.4, 82.6, 20.1, 90.8, 76.1, 30.92, 46.8, 67.1, 79.9"

#put your code below
```

שאלה 2 (20%)

ענו על כל שאלה באמצעות קובץ תכנית בן עד 15 שורות, והגש את הקובץ למערכת המטלות.

א. צור מחלקה בשם AppleBasket שהבנאי שלה מקבל שני ארגומנטים חיצוניים: מחרוזת המייצגת צבע, ומספר המייצג כמות. הבנאי צריך לאתחל שני משתני מופע: apple_color ו- apple_quantity. כתוב שיטה הנקראת increase המגדילה את הכמות באותו מופע ב- 1 בכל הפעלה. כמו כן יש לכתוב שיטה בשם __str__ למחלקה זו המחזירה מחרוזת בפורמט:

"A basket of [כמות] [צבע] apples."
Example1: A basket of 4 red apples.
Example2: A basket of 50 blue apples.

לאחר מכן יש ליצור שני מופעים ולהדפיס את תוכנם כך שיופיעו שתי הדוגמאות, בלי לקרוא מפורשות לאף שיטה. את התכנית יש להגיש בקובץ fruit.py .

ב. הגדר מחלקה שנקראת BankAccount המקבלת את שם החשבון המיועד בתור מחרוזת ומספר שלם המייצג של סכום היתרה בחשבון. הבנאי צריך לאתחל שני משתני מופע: name ו-amt בהתאמה. הוסף שיטה ליצירת מחרוזת כך שכאשר מדפיסים מופע של BankAccount, תתקבל ההודעה:

"Your account, [שם], has [יתרה] dollars."

צור מופע של המחלקה בשם Bob עם יתרה 100 ושמור אותו במשתנה t1, לאחר מכן הפעל עליו print. את הקובץ יש להגיש בשם bank.py .

שאלה 3 (60%)

א. בשאלה זו יש לבנות את המחלקות הבאות, עם בנאי מתאים לכל מחלקה:

- משתמשים, להם יש שם ומקצוע.
- מחלקה יורשת לפי מקצועות שונים: מהנדסים, טכנאים, ספרים, פוליטיקאים.
- כמו כן יש לבנות מחלקות לסוגי מהנדסים: מהנדסי חשמל, מהנדסי מחשבים, מהנדסי מכונות.
- התכנית תקבל קלט מהמשתמשים בזמן ריצה, שיאפשר לו להוסיף מחלקות נוספות לתכנית בזמן ריצה, תוך קבלת המידע הבא מהמשתמש: שם המחלקה החדש, שם משתנה חדש למחלקה, שם שיטה חדשה למחלקה והיא תיווצר בזמן ריצה.
- המשתמש יוכל להגדיר שם של מחלקה אם, כך שהמחלקה החדשה תוגדר כירושה ממחלקה זו.
- את התכנית יש להגיש בקובץ pros.py .

ב. כתבו תכנית המקבלת שם של קובץ פייתון המכיל מחלקה (לדוגמא, קבצים שכתבת לשאלות הקודמות בממ"ן זה), ושורת קוד בפייתון, מוסיפה לכל השיטות במחלקה את הקוד. נסו את התכנית על שורת הקוד print("Hello") והפעילו את שיטות המחלקה. את התכנית יש להגיש בקובץ meta.py .

-

בהצלחה!

מטלת מנחה (ממ"ן) 14

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה: יחידה 5 - תקשורת

מספר השאלות: 2 **משקל המטלה:** 4

סמסטר: 2021א **מועד אחרון להגשה:** 3.1.2021

בתרגיל זה נממש תוכנת **שרת** לגיבוי ואחזור קבצים ותוכנת **לקוח** שתעבוד מולו.
השרת יכתב בשפת C++ והלקוח בשפת python.

שרת (50%)

השרת יאפשר לכל לקוח לשלוח אליו קבצים לגיבוי ולשלוף את הקבצים האלו במועד מאוחר יותר.

מאפייני השרת:

א. השרת יתמוך בפרוטוקול חסר מצב (stateless)[2], כלומר, לא ישמור מידע בין בקשה לבקשה (כל בקשה עומדת בפני עצמה).
ב. השרת יתמוך בריבוי משתמשים ע"י תהליכונים (threads)

אופן הפעולה של השרת:

1. בלולאה אין סופית: ממתין לבקשות
2. בעת קבלת בקשה, יוצר thread חדש ומפענח את הבקשה לפי הפרוטוקול הנתון
3. ממשיך לפעול לפי הבקשה שהתקבלה:
   a. בקשה לשמירת קובץ לגיבוי:
   קבצים הנשלחים ע"י הלקוח ישמרו לתוך תיקיה יעודית של השרת, לכל משתמש תהיה תת-תיקיה ובתוכה הקבצים של אותו משתמש.
   לדוגמא: עבור לקוח מספר 1234 וקובץ בשם mmn14.pdf השרת ישמור את הקובץ בנתיב:
   c:\backupsvr\1234\mmn14.pdf
   b. בקשה למחיקת קובץ:
   מוחק את הקובץ הקיים.
   c. בקשה לרשימת הקבצים הקיימים:
   השרת יצור קובץ טקסט המכיל את רשימת הקבצים עבור לקוח זה.
   שם קובץ הטקסט יהיה אוסף תווים רנדומלי באורך 32 תווים (אותיות גדולות, קטנות באנגלית ומספרים)
   d. בקשה לאחזור קובץ:
   השרת ישלח כתשובה ללקוח את הקובץ המבוקש
4. אחרי הצלחה השרת יחזיר סטטוס הצלחה בהתאם לפרוטוקול
בכל מצב של שגיאה, השרת יחזיר סטטוס שגיאה בהתאם לפרוטוקול

[2] קראו כאן על פרוטוקול חסר מצב: https://en.wikipedia.org/wiki/Stateless_protocol

לקוח (50%)

הלקוח יעבוד מול השרת בהתאם לפרוטוקול.

בתחילת הריצה כל לקוח ייצר מספר אקראי ייחודי בגודל 4 בתים. מספר זה ישמש בכל הבקשות שיישלחו לשרת.

כתובת השרת והפורט יקראו <u>מתוך קובץ</u> בצורה הבאה:

- שם הקובץ: server.info
- מיקום הקובץ: באותה תיקיה של קובץ פייתון הראשי
- תוכן הקובץ: כתובת IP + נקודותיים + מספר פורט
  לדוגמא:
  127.0.0.1:1234

שמות הקבצים לגיבוי ואחזור יקראו מתוך קובץ בצורה הבאה:

- שם הקובץ: backup.info
- מיקום הקובץ: באותה תיקיה של קובץ פייתון הראשי
- תוכן הקובץ: שמות קבצים בלבד ללא נתיב (הקבצים יהיה באותה תיקיה של קובץ פייתון הראשי).
  לדוגמא:
  mmn14.pdf
  terminator2.avi

כך תראה תיקיה לדוגמא:

```
C:\openu\mmn14>dir /b
mmn14client.py
backup.info
mmn14.pdf
server.info
terminator2.avi

C:\openu\mmn14>type server.info
127.0.0.1:1234
C:\openu\mmn14>type backup.info
mmn14.pdf
terminator2.avi

C:\openu\mmn14>
```

אופן פעולת הלקוח:

1. **יוצר מספר אקראי ייחודי בגודל 4 בתים**

2. **קורא את כתובת השרת והפורט מתוך קובץ server.info**
3. **קורא את שמות הקבצים לגיבוי מתוך קובץ backup.info**
4. שולח בקשה לשרת לקבל את רשימת הקבצים הקיימים בגיבוי
- שרת מחזיר תשובה, יש להציג על המסך את רשימת הקבצים או את הודעת השגיאה שהתקבלה
5. שולח בקשה לשרת לשמירת הקובץ **הראשון** המופיע ב- backup.info
- שרת מחזיר תשובה, יש להציג על המסך את התשובה שהתקבלה (כולל שם הקובץ)
6. שולח בקשה לשמירת הקובץ **השני** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך
7. שולח בקשה לשרת לקבל את רשימת הקבצים הקיימים בגיבוי
- הדפסה של תשובת השרת למסך
8. שולח בקשה לאחזור הקובץ **הראשון** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך ושמירת הקובץ על הדיסק (לצד קובץ פייתון, בשם tmp)
9. שולח בקשה למחיקת הקובץ **הראשון** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך
10. שולח בקשה לאחזור הקובץ **הראשון** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך
11. סיום ויציאה

פרוטוקול התקשורת

עליכם לממש את הפרוטוקול הנתון מעל TCP.

כל השדות המספריים חייבים להיות עם ערכים גדולים מאפס (**unsigned**) ומיוצגים כ- **little endian**

בקשה:

Request	**שדה**	**גודל**	**משמעות**
כותרת (Header)	user id	4 בתים	מייצג את המשתמש
	version	בית	מספר גירסת לקוח
	op	בית	קוד בקשה
	name_len	2 בתים	אורך שם הקובץ
	filename	משתנה	שם הקובץ (ascii) **לא כולל תו מסיים** (null terminated)
תוכן (payload)	size	4 בתים	גודל הקובץ שנשלח
	Payload	משתנה	תוכן הקובץ (בינארי!)

בקשות אפשריות:

Op	**משמעות**	**הערות**
100	שמירה של קובץ לגיבוי	כל השדות מלאים
200	בקשה לאחזור קובץ	שדות size ו- payload לא קיימים
201	בקשה למחיקת קובץ	שדות size ו- payload לא קיימים
202	בקשה לרשימת כל	שדות name_len, filename, size, payload לא

	הקבצים של הלקוח	קיימים

תשובה:

Response	**שדה**	**גודל**	**משמעות**
כותרת (Header)	version	בית	מספר גירסת שרת
	status	2 בתים	סטטוס הבקשה
	name_len	2 בתים	אורך שם הקובץ
	filename	משתנה	שם הקובץ (ascii) **לא כולל תו מסיים** (null terminated)
תוכן (payload)	size	4 בתים	גודל הקובץ שנשלח
	Payload	משתנה	תוכן הקובץ (בינארי!)

תשובות אפשריות:

Status	**משמעות**	**הערות**
210	הצלחה	הקובץ נמצא ושוחזר. כל השדות מלאים
211	הצלחה	רשימת כל הקבצים חזרה ללקוח. כל השדות מלאים
1001	שגיאה	קובץ לא קיים. שדה size ו- payload לא קיימים
1002	שגיאה	אין קבצים על השרת ללקוח זה. רק שדות version ו- status קיימים
1003	שגיאה	שגיאה כללית. בעיה עם השרת רק שדות version ו- status קיימים

זיכרו! הפרוטוקול הוא בינארי.

כך תיראה לדוגמא בקשה לגיבוי קובץ:

offset					
0	1234	1	100	9	mmn14.pdf
17	29189	25 50 44 46 2D 31 2E 36 ...			

שימו לב!

הפרוטוקול מחייב ולא ניתן לעשות בו שינויים. כפועל יוצא, כל שרת ולקוח המממשים את הפרוטוקול יכולים לעבוד אחד מול השני.

דגשים לקוד שרת:

1. ממשו את התוכנה לפי עקרונות תכנות מונחה עצמים
2. מומלץ (אבל לא חובה) לעשות שימוש בספריות STL
3. ניתן ורצוי להשתמש ביכולות C++11 (לדוגמא פונקציות מסוג למדה, שימוש ב- auto וכו'..).
4. למימוש התקשורת עשו שימוש ב- winsock או בספרית boost
5. שימו לב לייצוג ערכים בזיכרון כ- little-endian או big-endian
6. לקוח יכול לשלוח קובץ בגודל דינמי **גדול**. חשבו על הדרך הנכונה ביותר לקבל כמות מידע גדולה מהלקוח.
7. הקפידו על תיעוד של הקוד (comments)
8. תנו שמות משמעותיים למשתנים, פונקציות ומחלקות. המנעו ממספרי קסם!
9. **אבטחת מידע**

חישבו לאורך כל הדרך על אבטחת מידע. האם בדקתם את הקלט? איך נעשה שימוש בזיכרון דינמי? האם מתבצעת המרת טיפוסים (casting) וכו'..
האם ואיך אפשר לתקוף את השרת? האם השרת יכול לתקוף את הלקוח?

דגשים לקוד לקוח:

1. השתמשו בפייתון **גירסה 3**
2. ממשו את התוכנה לפי עקרונות תכנות מונחה עצמים
3. עשו שימוש בספריות פייתון הסטנדרטיות
4. תוכלו להעזר בספריה **struct** על מנת לעבוד עם נתוני התקשורת בנוחות (בקשות/תשובות)
5. שימו לב לייצוג ערכים בזיכרון כ- little-endian או big-endian
6. השרת מאפשר קבלת קובץ בגודל דינמי **גדול**. חשבו על הדרך הנכונה ביותר לשלוח כמות מידע גדולה לשרת
7. הקפידו על תיעוד של הקוד (comments)
8. **אבטחת מידע**

האם תוכלו לתקף את השרת בצורה כלשהי? האם השרת יכול לתקוף את הלקוח?

הגשה:

5. שרת

א. עליכם להגיש רק את קבצי הקוד (כלומר קבצי h. ו- cpp.).
שימו לב! על התוכנית להתקמפל ולרוץ בצורה תקינה (ללא צורך בתוספות קבצים ללא קריסות)

ב. עבודתכם תיבדק במערכת הפעלה חלונות, באמצעות Visual Studio ולכן מומלץ לעבוד עם סביבה זו.

6. לקוח

א. עליכם להגיש רק את קבצי הקוד (כלומר קבצי py.).
שימו לב! על התוכנית לרוץ בצורה תקינה (ללא צורך בתוספות קבצים, ללא קריסות)

ב. יש לכלול פונקציה ראשית בשם **main**. פונקציה זו תהיה הפונקציה הראשית של תוכנית הלקוח והיא תעבוד לפי אופן פעולת הלקוח המוצג לעיל.
טיפ:
תוכלו להשתמש במנגנון הבא כדי לאפשר עבודה אינטראקטיבית **וגם** הרצה של הקוד

```
if __name__ == "__main__":
```

בהצלחה!