



בס"ד

שנה"ל תשפ"א, סמסטר ב, מועד א'
שאלון בחינה בקורס: מבוא להנדסת תוכנה
מספר קורס: 151050

<p>מס' תלמיד: _____</p> <p>קמפוס: _____</p> <p>למילוי ע"י הסטודנט</p>

- ❖ שם המרצה: מר גיא יחזקאל, מר דן זילברשטיין
- ❖ תאריך הבחינה: 4/07/2021, כ"ד תמוז, ה'תשפ"א
- ❖ משך הבחינה (בדקות): 150
- ❖ יש לקחת את הבחינה עם סיומה לצורך ערעורים
- ❖ חומר עזר מותר לשימוש: כל חומר עזר אסור לשימוש
- ❖ מחשבון: אסור לשימוש
- ❖ המבחן כולל 30 שאלות נכון \ לא נכון – סה"כ 60 נק', ו-10 שאלות בחירה (אמריקאיות) – סה"כ 40 נק'. יש לענות על כל השאלות.
- נבחן/ת יקר/ה,
- את התשובות יש לסמן בדף התשובות בלבד, השאלון ודפי הטיוטה לא ייבדקו.
- יש לרשום ולסמן במקומות המיועדים לכך בדף התשובות את מספר התלמיד ומספר המבחן. אין אפשרות לבדוק מבחן אשר נתונים אלה אינם מסומנים באופן מלא ונכון.
- אין לכתוב שום דבר נוסף על דף התשובות, רישומים נוספים פוסלים את הטופס לבדיקה.
- נוהל הבחינות של המרכז האקדמי לב מחייב אותך, באחריותך לקוראו ולהכירו - בחינה עלולה להיפסל על כל חריגה מהנוהל.

בהצלחה רבה !

5 שאלות נכון \ לא נכון. 10 נקודות. 2 נקודות כל שאלה

תאורת פלורסנט (**FluorescentLight**) היא תאורה בצורת גליל המאירה באור שווה מכל הנקודות על הגליל ובכל הכיוונים, בהתאם למרחק מהמנורה - בדומה להתפשטות האור ממקור תאורה נקודתי (**PointLight**). התאורה מכילה נקודת המייצגת את מרכז הגליל.

שאלה מספר 1:

יהיה זה נכון מבחינה עיצובית שהמחלקה **FluorescentLight** תממש את המחלקה **LightSource** ותו לא (מבחינת ירושה ומימוש ממשקים בלבד)

- א. לא נכון
- ב. נכון

שאלה מספר 2:

יהיה זה נכון מבחינה עיצובית שהמחלקה **FluorescentLight** תירש מהמחלקה **PointLight** ותו לא (מבחינת ירושה ומימוש ממשקים בלבד)

- א. נכון
- ב. לא נכון

שאלה מספר 3:

ללא שיפורי התמונה של **Soft Shadows** (צללים רכים) אין הבדל בין המחלקה **FluorescentLight** למחלקה **PointLight**

- א. נכון
- ב. לא נכון

שאלה מספר 4:

במידה ונבצע את שיפורי התמונה של **Soft Shadows** (צללים רכים), נרצה לתת אזור מטרה ליצירת אלומת קרניים. יהיה זה נכון מבחינה עיצובית ומספיק מבחינה פונקציונאלית שהמחלקה **FluorescentLight** תכיל אובייקט של המחלקה **Cylinder**

- א. נכון
- ב. לא נכון

שאלה מספר 5:

במידה ונבצע את שיפורי התמונה של **Soft Shadows** (צללים רכים), נרצה לתת אזור מטרה ליצירת אלומת קרניים. יהיה זה נכון מבחינה עיצובית ומספיק מבחינה פונקציונאלית שהמחלקה **FluorescentLight** תכיל אובייקט של המחלקה **Polygon**

- א. לא נכון
- ב. נכון

5 שאלות נכון \ לא נכון. 10 נקודות. 2 נקודות כל שאלה בהקשר של בניית תמונה שיפור תמונה וייעול בפריימוורק שבנינו:

שאלה מספר 6:

בשיטה של שיפור הביצועים על ידי **Regular Grid**: יצירת קופסא לכל הסצנה וחלוקה לנפחים שווים - ווקסלים (**Voxel**) - תהיה תמיד יעילה ככל שנגדיל את מספר הווקסלים.

- א. לא נכון
- ב. נכון

שאלה מספר 7:

בפיתוח שיפור התמונה **Anti-Aliasing** יהיה זה נכון מבחינת פיתוח מונחה אחריות (**Responsibility Driven Design - RDD**) לתת למחלקה **Camera** את פונקציית יצירת אלומות הקרניים.

- א. נכון
- ב. לא נכון

שאלה מספר 8:

בהקשר של שימוש בגופים תוחמים (**Conservative Bounding Region – CBR**) – השימוש בשיטה של קופסא תוחמת עם מקצועות מקבילים לצירים (**Axis Aligned bounding box – AABB**) – אמנם יכולה להוביל לתוצאות חיוביות מיותרות מבדיקת חיתוך, אך היא מהירה יותר לבדיקת חיתוכים מרוב סוגי גופים תוחמים אחרים

- א. נכון
- ב. לא נכון

שאלה מספר 9:

עבור שיפור התמונה של שקיפות והשתקפות (**Reflection & Refraction**) הפכנו את הפונקציה **CalcColor** לרקורסיבית. על מנת למנוע חישובי השתקפויות/שקיפויות מרובות מידי הוספנו שני תנאי עצירה. אחד מהם הוא על ידי שימוש במשתנה **MIN_CALC_COLOR_K**. השימוש במשתנה זה היה במטרה למנוע חישובים שלא ישפיעו רבות על הצבע הסופי בתמונה

- א. נכון
- ב. לא נכון

שאלה מספר 10:

מודל ה-**RGB** של הצבעים נלמד מחיישני המצלמה הצבעונית של ימינו המסוגלים לקלוט שלושה גוונים של צבעים

- א. לא נכון
- ב. נכון

5 שאלות נכון \ לא נכון. 10 נקודות. 2 נקודות כל שאלה בהקשר של עקרונות פיתוח, ארכיטקטורה ותבניות עיצוב תכנה.

שאלה מספר 11:

אנטי-תבנית (Anti-Pattern) מסוג שדון (Poltergeist) אופיינית יותר לשיטת Agile מאחר ועובדים בסבבים קצרים

- א. נכון
- ב. לא נכון

שאלה מספר 12:

שאלה 12 : בפיתוח בשיטה מונחית בדיקות TDD : בדיקות אינטגרציה (Integration Tests) מתבצעות רק לאחר סיום כל בדיקות הפיתוח ה-Unit Tests

- א. לא נכון
- ב. נכון

שאלה מספר 13:

ניתן להשתמש בתבנית העיצוב Wrapper Pattern גם בעזרת ירושה

- א. נכון
- ב. לא נכון

שאלה מספר 14:

בירושה ופולימורפיזם משתמשים בדריסה (override) עם האצלה (delegation) על מנת להפעיל את הפונקציה של מחלקת האב כאשר לא נדרש מימוש ייחודי במחלקת הבן

- א. לא נכון
- ב. נכון

שאלה מספר 15:

ביצירת המחלקה סצנה (Scene) השתמשנו בתבנית Builder Pattern. ראינו שתי גישות לשימוש בתבנית : כתיבת מחלקה פנימית שתיצור את המחלקה Scene ושינוי של המחלקה Scene שתממש את התבנית של Builder Pattern בצורה חלקית. ההבדל בין שתי הגישות הוא שבשיטת מחלקה פנימית ניתן לכתוב פונקציה הבודקת את תקינות והתאמה של כל נתוני האובייקט הנבנה

- א. נכון
- ב. לא נכון

5 שאלות נכון \ לא נכון. 10 נקודות. 2 נקודות כל שאלה בהקשר לנושאים השונים שנלמדו בקורס.

שאלה מספר 16:

שתמש בהכלה כאשר נרצה את רוב הפונקציונאליות של מחלקה קיימת אך המחלקה החדשה היא אינה מעין "סוג של" מחלקה קיימת

- א. נכון
- ב. לא נכון

שאלה מספר 17:

השתמשנו במחלקה **GeoPoint** אפילו שזה לכאורה נוגד את חוק הדמטר (**Law Of Demeter – LOD**) כדי שבסריקת קרניים נוכל לקבל נקודת חיתוך יחד עם הגוף הגאומטרי שנחתך, אך בנינו את המחלקה כמבנה נתונים פשוט (שטוח) (**Plain Data Structure – PDS**) על מנת שחוק הדמטר לא יגביל אותנו מגישה לאיברי המחלקה

- א. נכון
- ב. לא נכון

שאלה מספר 18:

ביצירת מחלקת **Color** השתמשנו בתבנית עיצוב **Wrapper Pattern** ובהאצלה (**Delegation**) לבנאי מחלקת **Color** של פריימוורק של **java** על מנת ליצור צבע סופי לפיקסל בתמונה.

- א. נכון
- ב. לא נכון

שאלה מספר 19:

האתגר בלמידתו של מרחב הבעיה (**Problem Domain**) שלמרות שהוא מוגדר בצורה מוחלטת, עדיין התהליכים בו מגוונים ורבים

- א. לא נכון
- ב. נכון

שאלה מספר 20:

תפקיד ה-**Acceptance Tests** (בדיקות קבלה) הן בדיקות סופיות של אנשי הפיתוח לפני מסירת המערכת ללקוח

- א. נכון
- ב. לא נכון

10 שאלות נכון \ לא נכון. 20 נקודות. 2 נקודות כל שאלה בהקשר של שיטות ותהליכים בפיתוח תכנה.

שאלה מספר 21:

חוקי להמן מתאימים רק למוצר מותאם לקוח ולא למוצר גנרי

- א. לא נכון
- ב. נכון

שאלה מספר 22:

היתרון בשיטת מפל המים האינקרמנטלי (Incremental Waterfall Model) לעומת שיטת מפל המים הקלאסי הוא ששלב ה-Design הוא גם חלק מהאיטרציה

- א. לא נכון
- ב. נכון

שאלה מספר 23:

אחד מתפקידי מהנדס תוכנה הוא לתרגם חלק מתאים מדרישות לא פונקציונאליות לדרישות פונקציונאליות

- א. נכון
- ב. לא נכון

שאלה מספר 24:

בנושא של מהימנות (Dependability) של המערכת בדגש על מאפיין של אבטחה (Security): כאשר מזהים את הנכסים שיכולים להיות מטרה לתקיפה (Asset Identification) יש צורך להגן על כל הנכסים שזיהינו.

- א. לא נכון
- ב. נכון

שאלה מספר 25:

במערכת מבוזרת לניהול גרסאות בהכרח לא יכול להיות מאגר מרכזי

- א. לא נכון
- ב. נכון

שאלה מספר 26:

שימוש חוזר בתכנה (Software Reuse) מאפשר למנהלי הפרוייקט להעריך נכון יותר את זמני פיתוח הפרוייקט

- א. נכון
- ב. לא נכון

שאלה מספר 27:

חיזוי שינויים העתידיים להתרחש בתוכנה הוא מטרה המתאימה לפרוייקטים מורכבים שלא ניתן לשנות או קשה מאוד לשנות אותם אחרי סיום הפיתוח

- א. נכון
- ב. לא נכון

שאלה מספר 28:

יש משמעות רבה בשימוש באבטיפוס (Prototype) למערכות מסוג P-Type

א. לא נכון

ב. נכון

שאלה מספר 29:

Main Line הינו שחרור גירסא ללקוח על בסיס Base Line מסוים

א. לא נכון

ב. נכון

שאלה מספר 30:

מערכת זמן אמת נותנת עדיפות לעיבוד גירויים מזדמנים על פני גירויים מחזוריים

א. לא נכון

ב. נכון

10 שאלות אמריקאיות (בחר תשובה נכונה). 40 נקודות. 4 נקודות כל שאלה

שאלה מספר 31:

בהקשר להגדרות של ריחות הקוד (code smells), בחר בתשובה הנכונה.

- א. שמירה על עקרון (Law Of Demeter – LOD) תתרום ל-**loose coupling** (קישוריות חלשה) בין מחלקות
- ב. שמירה על עקרון ה (Don't Repeat Yourself – DRY) תמנע מהתכונה סרחון של שבריריות (**fragility**)
- ג. פיתוח מונחה אחראיות (**Responsibility Driven Design – RDD**) יפחית את סרחון ה-**immobility** (חוסר ניידות)
- ד. כאשר הקוד הינו **rigid** (נוקשה), הסיכוי לכך שהוא יהפוך ל-**fragile** (שברירי) נמוך
- ה. כאשר קוד כתוב בצורה שאינה ברורה הרי הוא סובל מחוסר ניידות (**immobility**)

שאלה מספר 32:

בהקשר לעקרונות פיתוח תוכנה **Agile**, בחר בתשובה הנכונה.

- א. שיטת **Agile** הצורך לבצע **refactoring** (ארגון מחדש) של הקוד הוא גדול יותר מאשר בשיטת מפל המים (**Waterfall**)
- ב. בשיטת **Agile** אין צורך בתיעוד, בתרשימים, ומסמכים כלל מאחר והקוד וההערות בקוד הן התיעוד של התוכנה
- ג. בשיטת **Agile** חייבים להשקיע מראש מאמץ של תכנון מפורט העוסק במניעת פגמים ובאגים - על מנת להפחית את הסיכונים בפיתוח המערכת
- ד. בשיטת **Agile** מבוסס בעיקרו על פיתוחי אבטיפוס (**prototype**)
- ה. בשיטת **Agile** חובה להשתמש בפיתוח מונחה בדיקות (**Test Driven Design – TDD**)

שאלה מספר 33:

בהקשר למערכת לניהול גרסאות **git**, בחר בתשובה הנכונה.

- א. על מנת ליצור ענף (**branch**) חדש ל-**codeline** ולהמשיך לעדכן בו שינויים יש לבצע את רצף הפקודות הבאות (משמאל לימין): **branch, checkout, commit, commit**
- ב. הדרך המהירה ביותר למניעת התנגשויות (**conflict**) ב-**git** היא פעולת **rebase**
- ג. ההבדל בין **git** לשאר מערכות לניהול גרסאות הוא ש-**git** שומר את כל הקבצים מחדש לכל גרסה וגרסה ושאר מערכות ניהול שומרות רק את השינויים (דלתא) בין הגרסאות
- ד. בפקודת **commit** בכל מקרה כל הקבצים המנוהלים ששונו מאז ה-**commit** האחרון עוברים למאגר (**repository**)
- ה. פקודת **reset** מבצעת יצירת **commit** חדש עם תוכן הגרסה הקודמת כך שניתן לחזור אחר כך גם לגרסה הנוכחית

שאלה מספר 34:

הקשר לממשק סימון (**Marker Interface**) שנלמד בקורס, בחר בתשובה הנכונה.

- א. תבנית ממשק סימון (**Marker Interface**) לא דורשת פונקציונאליות נוספת על מנת לממש אותה
- ב. ההחלטה לא להשתמש בתבנית סימון נבעה מהסיבה שהיא מוגדרת לכתחילה כאנטי-תבנית (**anti-pattern**)
- ג. ההחלטה לא להשתמש בתבנית סימון נבעה מהסיבה שהיא התאימה רק לצורות שטוחות
- ד. ההחלטה לא להשתמש בתבנית סימון נבעה מהסיבה שהיא דרשה לבצע **refactor** (ארגון מחדש) רב של הקוד
- ה. המחשבה להשתמש בתבנית סימון נבעה בגלל בעיות מיקום הגיאומטריות בסצנה שיצרו הצללה חלקית בלבד

שאלה מספר 35:

בהקשר של מערכות זמן אמת, בחר בתשובה הנכונה.

- א. במערכות זמן אמת יש שני סוגים של גירויים: גירויים מחזוריים (המגיעים בקצב קבוע) וגירויים מזדמנים (המגיעים בעקבות אירוע שמתרחש)
- ב. כל מערכות זמן אמת חייבות לעמוד ב-**deadline** של כל הגירויים שמתקבלים על ידי החיישנים
- ג. מערכות אינטראקטיביות (**interactive**) הן מערכות זמן אמת לכל הדעות
- ד. ארכיטקטורה מערכת זמן אמת מסוג **Observe and React** מתאימה למערכות שצריכות לעקוב גם אחרי המצב של המפעילים
- ה. שלושת תבניות העיצוב שלמדנו עבור מערכות זמן אמת הינן תבניות התנהגותיות (**Behavioral**)

שאלה מספר 36:

בהקשר של מודלים של התפשטות האור והחזרת האור של פונג (**Phong Reflection Model**) בחר בתשובה הנכונה.

- א. המשתנה n_{sh} בנוסחה נועד להגדיל את ריכוז השפעת הרכיב הספקולרי (**Specular**) בהחזרת אור של מקור תאורה
- ב. החזרת האור במודל פונג מתפרקת לארבעה רכיבים – ספקולרי (**Specular**), דיפוזיבי (**Diffuse**), שקיפות (**Reflection**) והשתקפות (**Refraction / Transparency**)
- ג. החזרת האור של תאורה נקודתית (**Point Light**) מהגאומטריה דועכת עם המרחק
- ד. מודל פונג (**Phong**) בנוי על בסיס מודל פיזיקלי של החזרת האור
- ה. על פי מודל פונג, היחס בין כל מרכיבי התאורה הינו הכפולתי (יש לכפול את תרומת ההארה של כל אחד מן מקורות האור לנקודה)

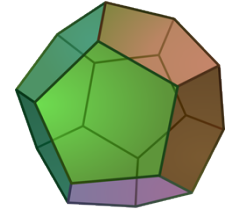
שאלה מספר 37:

מה מהבאים נכון לגבי פיתוח מונחה בדיקות **TDD**. בחר בתשובה הנכונה.

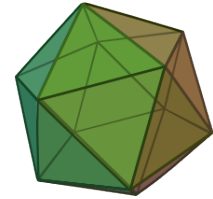
- א. פיתוח מונחה בדיקות מאפשר בניית בדיקות מפתח על פי דרישות המערכת תוך ניתוק התלות בין הבדיקות לבין הקוד של המימוש
- ב. פיתוח מונחה בדיקות תורם לכתיבת מקרי בדיקה יותר מקיפים
- ג. מטרת פיתוח מונחה בדיקות היא חלוקה יותר נכונה של מקרי הבדיקה על ידי שימוש במחלקות שקילות (**Equivalence Partitioning**) ובדיקות גבול (**Boundary Value Analysis**)
- ד. פיתוח מונחה בדיקות מאפשר תיעוד ברור יותר של הקוד וחיסכון במסמכי בדיקות גדולים
- ה. על ידי שימוש בפיתוח מונחה בדיקות מקבלים קוד יותר איכותי מאחר שהקוד תמיד עובר תהליך של ארגון מחדש (**refactoring**)

שאלה מספר 38:

נרצה להוסיף לסצנה צורות גיאומטריות עם פאות מסוגים שונים. לדוגמא הצורה "תריסרון משוכלל" (Regular Dodecahedron) הוא צורה בעלת 12 פאות שכל פאה היא מחומש משוכלל:



דוגמה נוספת היא הצורה "עשרימון משוכלל" (Icosahedron) המורכב מעשרים פאות שהן משולשים שווים צלעות:



לצורך זה ניצור מחלקה **Polyhedral** ונוסיף אותה לחבילת הגאומטריות. בחר בתשובה הנכונה.

- יהיה זה נכון מבחינה עיצובית ומספיק מבחינה פונקציונאלית שבמחלקה **Polyhedral** יהיה אוסף של אובייקטים מטיפוס המחלקה **Polygon**
- יהיה זה נכון מבחינה עיצובית ומספיק מבחינה פונקציונאלית שהמחלקה **Polyhedral** תירש מהמחלקה **Polygon**
- יהיה זה נכון מבחינה עיצובית ומספיק מבחינה פונקציונאלית שמחלקה **Polyhedral** תכיל את המחלקה **Triangle**
- יהיה זה נכון להשתמש בתבנית הבנאי (**Builder Pattern**) לצורך עיצוב המחלקה **Polyhedral**
- יהיה זה נכון להשתמש בתבנית **Composite** בעיצוב המחלקה **Polyhedral**

שאלה מספר 39:

איזו פונקציה לא עברה שינוי ולא נוספה כתוצאה מהוספת פונקציונאליות של שקיפות (**Refraction**) והשתקפות (**Reflection**) (כולל ארגון קוד מחדש לאחר מימוש שקיפות והשתקפות). בחר בתשובה הנכונה.

- calcLocalEffects**
- unshaded**
- calcColor**
- traceRay**
- calcGlobalEffects**

שאלה מספר 40:

בהקשר של מודל השלבים (Staged Model) של מחזור החיים של מערכות תוכנה (Software System Life Cycle) בחר בתשובה הנכונה.

- א. השלב השני - **Evolving** (התפתחות המערכת) הוא השלב האיטרטיבי המאפיין את ההתפתחות העיקרית של המערכת
- ב. בשלב ה-**Phase out** לא נותנים יותר תמיכה למוצר אלא מתקדמים לפתח את המוצרים הבאים בלבד
- ג. בשלב ה-**Servicing** עדיין ניתן להוסיף פונקציונאליות חדשה בתנאי שהיא יחסית קטנה
- ד. בשלב ה-**Close Out** אין יותר תיקונים של באגים במערכת אבל עדיין מספקים תמיכה ללקוחות
- ה. מודל השלבים מתאים יותר לפיתוח בשיטת מפל המים (Waterfall) מאשר **Agile**