



תואר ראשון בהנדסת מערכות-מידע ותוכנה

מס' קורס - 372-12104 - נושאים מתקדמים בתכנות

Advanced Topics in Programming

שעור: דר. דוד בן שמעון

סמסטר ב, שנה"ל - תש"פ 2020 – 2019

יום ג': קבוצה א 18:00 – 16:00 קבוצה ב – 20:00 – 18:00

מתרגלים: מר גיא שלזינגר, מר עידו סקזי

קבוצות תרגול - יום ה'

המטרה

מטרת הקורס היא הרחבת יכולות התכנות של הסטודנטים והיכרות מעמיקה עם נושאים מתקדמים בתחום התוכנה ועיצובה.

התוצר הסופי מהקורס הינו יכולת גבוהה יותר בעיצוב תוכנה מונחית עצמים, ידע מעמיק יותר בשפת Java, וידע נרחב בפתרונות עיצוביים לבעיות נפוצות בפיתוח ועיצוב תוכנה – design patterns.

קורסי קדם

מבוא לתכנות – 202-1-1041

מבני נתונים – 202-1-1051

מבנה הקורס

הקורס בנוי משעתיים הרצאה ושעתיים תרגול. בהרצאה יעברו התכנים באופן עיוני ותיאורטי ואילו בתרגול ייכנסו למעשי. ישנו פרויקט אחד גדול בקורס המחולק לשלושה אבני דרך ואשר

מלווה בהרצאות ובתרגולים כמובן. במסגרת הפרויקט, הסטודנטים נדרשים לכתוב תוכנה בשפה מונחית עצמים תוך הפרדה בין שכבות קוד ושימוש בתבניות עיצוב לפתרון בעיות נפוצות. במסגרת הקורס הסטודנטים ייחשפו למגוון רחב של אספקטים הקשורים לתכנות – תכנות מקבילי, גישה ועבודה עם בסיסי נתונים וקבצים, טיפול וניהול שגיאות, תיעוד, רשתות ותקשורת בין אפליקציות, מחרוזות ועוד.

דרישות מהתלמיד

הגשת כל אבני הדרך של הפרויקט וציון עובר במבחן.

אתר הקורס

<https://moodle2.bgu.ac.il/moodle/>

לאחר יועלו ההרצאות, התרגילים, הודעות, ציונים, חומר עזר וכו'. יש להתעדן שם לפני כל הרצאה.

הרכב הציון

75% - בחינה. 25% - תרגילים. חובה לעבור את הבחינה.

אין בחנים.



נושאים רלוונטיים לקורס (יתכנו שינויים קלים)

#Lecture	Subjects
Lecture 1+2	<ul style="list-style-type: none"> • Introduction to java <ul style="list-style-type: none"> ◦ Java architecture ◦ Important concepts ◦ Arrays ◦ Strings ◦ Parameter Passing • Class Diagram as part of UML • Classes and Objects • Object oriented Programming <ul style="list-style-type: none"> ◦ Encapsulation ◦ Inheritance ◦ Polymorphism • Delegation Vs. Inheritance • Abstract Classes • Interfaces
Lecture 3	<ul style="list-style-type: none"> • Design Principles <ul style="list-style-type: none"> ◦ Code smells ◦ Solid • Design Patterns <ul style="list-style-type: none"> ◦ Definition ◦ Type • Bridge Design Pattern • The maze problem
Lecture 4	<ul style="list-style-type: none"> • From story to Object Oriented Programming
	<ul style="list-style-type: none"> •
Lecture 5	<ul style="list-style-type: none"> • Java Generics • Java Containers • Factory Design Pattern
Lecture 6	<ul style="list-style-type: none"> • Networking – Sockets • Strategy Design Pattern
Lecture 7	<ul style="list-style-type: none"> • Threads • Processes • Multithreaded client-server
Lecture 8	<ul style="list-style-type: none"> • Concurrent Design Pattern • Thread pool • Server-Example (36) using thread pool
Lecture 9	<ul style="list-style-type: none"> • <u>Event Driven Programming –</u> <ul style="list-style-type: none"> ◦ Observer Design Pattern ◦ JavaFX

Lecture 10	<ul style="list-style-type: none">• Accessing data<ul style="list-style-type: none">○ Data Bases○ Files• Singleton design pattern
Lecture 11	<ul style="list-style-type: none">•
Lecture 12	<ul style="list-style-type: none">• Architectural Patterns – Model View Controller
Lecture 13	<ul style="list-style-type: none">• Virtual Money<ul style="list-style-type: none">○ Transactions○ Block chain○ Mining○ Hashing

מקורות

- **Design Patterns: Elements of Reusable Object-Oriented Software** by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch
- **Head First Design Patterns** by Kathy Sierra, Bert Bates, Elisabeth Robson, Eric Freeman
- **Java: The Complete Reference, 9th Edition** by Herbert Schildt (Author)