**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Львівський національний університет імені Івана Франка**

**Факультет прикладної математики та інформатики**

**Кафедра прикладної математики**

**Затверджено**

На засіданні

кафедри теорії оптимальних процесів

факультету прикладної математики та інформатики

Львівського національного університету імені Івана Франка

(протокол № \_\_\_\_ від \_\_\_\_\_\_\_ 2020\_\_ р.)

Завідувач кафедри Шахно С.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Силабус з навчальної дисципліни**

**“Сучасні технології програмування”,**

**що викладається в межах ОПП Системний аналіз**

**другого (магістерського) рівня вищої освіти для здобувачів з спеціальності 124 – системний аналіз**

**Львів 2020 р.**

|  |  |
| --- | --- |
| **Назва дисципліни** | Сучасні технології програмування |
| **Адреса викладання дисципліни** | Головний корпус ЛНУ ім. І. Франка  м. Львів, вул. Університетська 1 |
| **Факультет та кафедра, за якою закріплена дисципліна** | Факультет прикладної математики та інформатики  Кафедра теорії оптимальних процесів |
| **Галузь знань, шифр та назва спеціальності** | 12 – інформаційні технології  124 – системний аналіз |
| **Викладачі дисципліни** | Голуб Богдан Михайлович, доцент кафедри теорії оптимальних процесів  Шунькін Юрій Валерійович, асистент кафедри теорії оптимальних процесів |
| **Контактна інформація викладачів** | [bohdan.holub@lnu.edu.ua](mailto:bohdan.holub@lnu.edu.ua); <https://ami.lnu.edu.ua/employee/holub>;  [yuriy.shunkin@lnu.edu.ua](mailto:yuriy.shunkin@lnu.edu.ua); <https://ami.lnu.edu.ua/employee/shunkin-yu-v>; Головний корпус ЛНУ ім. І. Франка, каб. 269.  м. Львів, вул. Університетська, 1 |
| **Консультації з питань навчання по дисципліні відбуваються** | Консультації в день проведення лекцій/практичних занять (за поперед­ньою домовленістю). |
| **Сторінка курсу** | <https://teams.microsoft.com/_#/school/conversations/%D0%97%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5?threadId=19:979a0bf5d8cb4a49a848aae28ab6ebc6@thread.tacv2&ctx=channel> |
| **Інформація про дисципліну** | Курс розроблено таким чином, щоб надати учасникам базові знання сучасних технологій програмування, перелік яких формується на основі рейтингу використання алгоритмічних мов |
| **Коротка анотація дисципліни** | Дисципліна “Сучасні технології програмування” є нормативною дис­ципліною зі спеціальності 124 – системний аналіз для освітньої програми Системний аналіз, яка викладається у 1-му семестрі в обсязі 2-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS). |
| **Мета та цілі дисципліни** | Метою вивчення нормативної дисципліни “Сучасні технології програмування” є надання студентам базових знань та практичних навичок використання сучасних технологій програмування. |
| **Література для вивчення дисципліни** | 1. Основи JavaScript / <https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics> 2. Руководство по языку программирования Java / <https://metanit.com/java/tutorial/> 3. Руководство по языку программирования Python / <https://metanit.com/python/tutorial/> 4. PHP Manual / <https://www.php.net/manual/en/index.php> 5. Б.М.Голуб. C#. Концепція та синтаксис. Навчальний посібник. – Львів: ВНТЛ, 2016. – 144с. 6. CSS first steps / <https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps> 7. Ruby – Вікіпідручник / <https://uk.wikibooks.org/wiki/Ruby> 8. Руководство по языку программирования Swift / <https://metanit.com/swift/tutorial/> 9. A Tour of Scala / <https://docs.scala-lang.org/tutorials/> 10. Интерпретатор команд Shell / <https://www.lissyara.su/doc/programming/shell/> 11. K.Seguin. The Little Go Book / <https://www.openmymind.net/assets/go/go.pdf> 12. Освоюємо R / <https://uk.wikibooks.org/wiki/%D0%9E%D1%81%D0%B2%D0%BE%D1%8E%D1%94%D0%BC%D0%BE_R> 13. TypeScript. Basic Types / <https://www.typescriptlang.org/docs/handbook/basic-types.html> 14. А.Попов. Введение в Windows PowerShell /  <https://andpop.ru/courses/winscript/books/posh_popov.pdf> 15. Perl Вступ / <http://docs.linux.org.ua/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/Perl/Perl_%D0%92%D1%81%D1%82%D1%83%D0%BF/> 16. А.Холомьёв. Учебник по Haskel. <https://anton-k.github.io/ru-haskell-book/files/ru-haskell-book.pdf> |
| **Обсяг курсу** | Загальний обсяг: 80 годин. Аудиторних занять: 48 год., з них 16 год. лек­цій та 32 години лабораторних робіт. Самостійної роботи: 32 год. |
| **Очікувані результати навчання** | Після завершення цього курсу студент буде :  Знати основи тенологій:  JavaScript, Java, Python, PHP, C#, CSS, Ruby, Swift, Scala, Shell, Go, R, TypeScript, PowerShell, Perl, Kotlin  Вміти: застосовувати набуті знання для розробки програмного забезпечення |
| **Ключові слова** | JavaScript, Java, Python, PHP, C#, CSS, Ruby, Swift, Scala, Shell, Go, R, TypeScript, PowerShell, Perl, Kotlin |
| **Формат курсу** | Очний, дистанційний  Проведення лекцій, лабораторних робіт і консультацій. |
| **Теми** | * JavaScript * Java * Python * PHP * C# * CSS * Ruby * Swift * Scala * Shell * Go * R * TypeScript * PowerShell * Perl * Kotlin |
| **Підсумковий контроль, форма** | Комбінований залік і екзамен у кінці семестру |
| **Пререквізити** | Для вивчення курсу студенти потребують базових знань з   * Програмування; |
| **Навчальні ме­тоди та тех­ніки, які будуть ви­користовува­тися під час викладання курсу** | Презентації, лекції  Індивідуальні завдання  Групові проекти, менторство |
| **Необхідне об­ладнання** | Комп’ютер та Internet-доступ до хмарного середовища (Amazon/Windows Asure/Google (App Engine)). |
| **Критерії оці­нювання (ок­ремо для кож­ного виду нав­чальної діяль­ності)** | Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:  • індивідуальні завдання: 50% семестрової оцінки; максимальна кількість балів 50  • екзамен: 50% семестрової оцінки; максимальна кількість балів 50  Підсумкова максимальна кількість балів 100.  **Лабораторні роботи:** Очікується, що студенти виконають 8 лабораторних завдань за вільним вибором 8 тем.  **Академічна доброчесність**: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади мож­ли­вої академічної недоброчесності. Виявлення ознак академічної недоб­ро­чесності в письмовій роботі студента є підставою для її незарахуван­ння викладачем, незалежно від масштабів плагіату чи обману.  **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та практичні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов’язані дотримуватися термінів виз­начених для виконання всіх видів письмових робіт та індивідуальних завдань, передбачених курсом.  **Література.** Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.  П**олітика виставлення балів.** Враховуються бали набрані при поточному тестуванні, самостійній роботі та бали підсумкового тестування. При цьому обов’язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнень на заняття; користування мобільним телефоном, планшетом чи іншими мо­більними пристроями під час заняття в цілях не пов’язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.  Жодні форми порушення академічної доброчесності не толеруються. |
| **Питання до заліку чи екзамену.** | JavaScript  Java  Python  PHP  C#  CSS  Ruby  Swift  Scala  Shell  Go  R  TypeScript  PowerShell  Perl  Kotlin |
| **Опитування** | Анкету-оцінку з метою оцінювання якості курсу буде надано по завер­шенню курсу. |