**Потоки** (лекції Мухи):

1. Що таке процес читання? запису?
2. Що таке потік?
3. Що таке буфер? Для чого він слугує?
4. У яких форматах відбувається обмін інформації в потоках? В чому вони різняться?
5. Що таке текстовий потік? Його особливості.
6. Що таке бінарний потік? Його особливості.
7. Що таке стандартні потоки? Приклади стандартних потік в ANSI C та C++?
8. Особливості сerr, clog?
9. Файловий ввід/вивід в С++ (бібліотеки, класи) ?
10. Класифікація потів інформації?
11. Коли відбувається відкриття файла?
12. Основні режими відкриття файлу в С++?
13. Чи можна комбінувати прапори відкриття файлу?
14. Параметри access?
15. Значення ofsteram, instream, fstream за замовчуванням?
16. Як перевірити що файл відкритий?
17. Як визначити кінець файлу?
18. Як закрити файл?
19. Як відкрити текстовий файл? Бінарний?
20. Зчитування строки з файла?
21. Форматування в потоках?
22. Робота з бінарними файлами?
23. Як визначити кількість успішно записаних символів в потоках С++?
24. Що відбувається при досягнені кінця файлу?
25. Чи можна використовувати get, put для бінарного введення/виведення?
26. Як організувати випадковий доступ до файлу?
27. Які бувають види напрямків в функція довільного доступу?
28. Як отримати поточне значення покажчика в файлі?

**Класи** (практика №1):

1. Що таке вбудована функція?
2. Що таке перегрузка функцій?
3. Правила перегрузки функції?
4. R-value ссилка?
5. Що таке ідеальна передача?
6. Що таке клас?
7. Що таке інкапсуляція?
8. Що включає в себе опис класу?
9. Рівні доступу до даних?
10. Навіщо захищені елементи?
11. Що не може бути елементом-даних в класі?
12. Коли необхідна одна копія даних (властивостей) і як цього добитися?
13. Коли мають задаватися начальні значення статичного елемента?
14. Що таке функції-елементи класу?
15. Чому елементи-дані мають бути завжди захищеними?
16. Як можна заборонити функції модифікацію об’єкту класу, що в неї передається.
17. Як можна дозволити змінювати дані класу, якщо сам об’єкт класу переданий в функцію як константний.
18. Чи можуть бути функції статичними?
19. Які є види конструкторів (4) ?
20. Які конструктори неявно створює компілятор?
21. Що таке конструктор за замовчуванням, коли він використовується (5)? Коли конструктор за замовчування не створюється компілятором?
22. Навіщо існує конструктор ініціалізації?
23. Коли використовується список ініціалізації?
24. Які правила користування конструктором (8) ?
25. Коли необхідно описувати свій конструктор (2) ?
26. Що таке деструктор?
27. Правила використання деструкторів (7) ?
28. Які є способи створення об’єктів, їх особливості (10)?
29. Що таке поверхневе копіювання та глибоке копіювання?

**Лекція\_01**:

1. Причина виникнення ООП?
2. Програма в термінах ООП?
3. Взаємодія об’єктів в ООП?
4. ООП?
5. Головна ідея ООП?
6. Фундаментальні поняття ООП?
7. Об’єкт?
8. Атрибут?
9. Метод?
10. Повідомлення?
11. Об’єкт (за структурою)?
12. Клас?
13. Об’єкт – екземпляр … ?
14. Інкапсуляція?
15. Протокол класу?
16. Наслідування?
17. Види ієрархій?
18. Види успадкування?
19. Абстрактний клас?
20. Поліморфізм?
21. Як відбувається реалізація поліморфізму?
22. Основні принципи ООП (3)?
23. Фундаментальні характеристики ООП (6)?
24. Етапи створення ООП програм?
25. ООА?
26. ООD?
27. OOP?

**Лекція\_02**:

1. Атрибути?
2. Методи?
3. РБНФ класу?
4. Чим є ідентифікатори елементів класу по відношенню до класу?
5. Чим можна ініціалізувати поля класу під час його створення?
6. Специфікатори доступу? Які вони є?
7. Якими є функції, що описані в тілі класу?
8. Конструктор?
9. Ініціалізація полів об’єкту без конструктора ініціалізації.
10. Доступ до членів класу(2)?
11. Властивості конструктора?
12. Властивості деструктора?
13. Коли необхідно використовувати конструктори?
14. Правила виклику конструкторів та деструкторів?
15. Порядок виклику конструкторів?
16. Порядок виклику деструкторів?