**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 потоків інформації (за напрямком обміну 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 виклику перегружених функцій?
4. R-value ссилка?
5. Що таке ідеальна передача?
6. Що включає в себе опис класу?
7. Рівні доступу до даних?
8. Навіщо захищені елементи?
9. Що не може бути елементом-даних в класі?
10. Коли необхідна одна копія даних (властивостей) і як цього добитися?
11. Коли мають задаватися начальні значення статичного елемента?
12. Що таке функції-елементи класу?
13. Чому елементи-дані мають бути завжди захищеними?
14. Як можна заборонити функції модифікацію об’єкту класу, що в неї передається.
15. Як можна дозволити змінювати дані класу, якщо сам об’єкт класу переданий в функцію як константний.
16. Чи можуть бути функції статичними?
17. Які є види конструкторів (5) ?
18. Які конструктори неявно створює компілятор?
19. Що таке конструктор за замовчуванням, коли він використовується (5)? Коли конструктор за замовчування не створюється компілятором?
20. Навіщо існує конструктор ініціалізації?
21. Коли використовується список ініціалізації?
22. Які правила користування конструктором (8) ?
23. Коли необхідно описувати свій конструктор копіювання (2) ?
24. Що таке деструктор?
25. Правила використання деструкторів (7) ?
26. Які є способи створення об’єктів, їх особливості (10)?
27. 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. Принципи ООП?
7. Фундаментальні поняття ООП?
8. Атрибут?
9. Метод?
10. Повідомлення?
11. Об’єкт?
12. Клас (через сукупність обєктів)?
13. Об’єкт – екземпляр … ?
14. Інкапсуляція?
15. Протокол класу?
16. Наслідування?
17. Види ієрархій?
18. Види успадкування?
19. Наслідування “is-a”?
20. Наслідування “part of”?
21. Базовий клас?
22. Абстрактний клас?
23. Що відрізняє ООП від об’єктної парадигми?
24. Поліморфізм?
25. Як відбувається реалізація поліморфізму?
26. Фундаментальні характеристики ООП (6)?
27. Етапи створення ООП програм?
28. ООА?
29. ООD?
30. OOP?

**Лекція\_02**:

1. РБНФ класу?
2. Що задає тіло класу?
3. Чим є ідентифікатори елементів класу по відношенню до класу?
4. Чи можна ініціалізувати поля класу під час його створення? Чому?
5. Чому елементом класу не може бути об’єкт даного класу?
6. Специфікатори доступу? Які вони є? Що задають та для чого служать?
7. Які є способи визначення функцій?
8. Якими є функції, що описані в тілі класу?
9. Від чого залежить спосіб ініціалізації об’єкту?
10. Конструктор?
11. Ініціалізація полів об’єкту без конструктора ініціалізації.
12. Доступ до членів класу(повне та скорочене ім’я)?
13. Від чого залежать способи доступу до членів класу?
14. Особливості конструторів (9)?
15. Конструктор за замовчуванням?
16. Види конструкторів за замовчуванням (3)?
17. Чи може клас містити декілька конструкторів за замовчуванням? Жодного?
18. Конструктор з параметрами?
19. Різновиди конструкторів з параметрами (3)?
20. Для чого не можна використовувати конструктори з параметрами (для ініціалізації полів…)?
21. Як ініціалізувати константні поля в класі за допомогою конструктора?
22. Особливості використання списку ініціалізації (прядок визначення)?
23. Конструктор копіювання?
24. Коли використовується конструктор копіювання (4)?
25. Поверхневе копіювання/глибоке копіювання?
26. Конструктор переміщення?
27. Конструктор перетворення?
28. Деструктор?
29. Особливості задання деструкторів (4)?
30. Особливості використання деструкторів (3)?
31. Коли необхідно явно викликати деструктор?
32. Особливості роботи з об’єктами, що містять динамічні атрибути?
33. Передача об’єктів функціям?
34. Механізм повернення обєктів за значенням?
35. Правила виклику конструкторів та деструкторів?
36. This?
37. Константні методи?
38. Особливості статичних членів класу?
39. Порядок виклику конструкторів?
40. Порядок виклику деструкторів?

**Лекція\_03**:

1. Перевизначення операцій?
2. Які оператори не можна перегризти?
3. Різниця між перегрузкою функції та перевизначенням операцій?
4. Які перевантаження функції надаються компілятором автоматично?
5. Особливості перевизначення унарних операторів?
6. Особливості перегрузки операторів ++/--?
7. Навіщо перегружати оператор “->”?
8. Особливості перевантаження бінарних операторів?
9. Який об’єкт маю повертати арифметичні оператори?
10. Особливості перевантаження логічних операторів?
11. Різниця між конструктором копіювання та оператором присвоєння?
12. Особливості перевантаження оператора “,”?
13. Перевизначення операторів new, delete, new [], delete [] та коли вона може використовуватися?
14. Перевизначення оператора «()» та його особливості?
15. Які оператори не можна перевизначати за допомогою дружніх функцій?
16. Як використовується перевизначення бінарних операторів за допомогою дружніх функцій?