**Lab- 1**

1. Ide – середовище для розробки програмного забезпечення.
2. Графічний інтерфейс .
3. Visual Studio, VS Code, Code Blocks.
4. Найзручніша IDE, не маю уявлення як її описувати.
5. Гарячі клавіші, завдяки яким зручніше працювати.
6. Їх дуже багато
7. Ctrl+f5.
8. Ctrl+F
9. F12
10. Project-> new project-> type: .net Framework (windows forms)
11. Button, radio button та їх багато всі важко назвати
12. Перетягнути з ToolBox
13. Натиснути на елементи, та подивитися дані в колонці (properties)

**Lab- 2**

1. Шаблон об’єкта
2. Екземпляр класу
3. Private, Public, Protected, internal, protected internal, protected internal.
4. Конструктор – це спеціалізований метод, який використовується при створенні екземляру класу. (Ну частково є методом).
5. Конструктор створює об’єкт і задає йому значення, Деструктор видаляє об’єкт.

**Lab- 3**

1) Шаблон об’єкта

1. Екземпляр класу
2. Private, Public, Protected, internal, protected internal, protected internal.
3. Конструктор – це спеціалізований метод, який використовується при створенні екземляру класу. (Ну частково є методом).
4. Конструктор створює об’єкт і задає йому значення, Деструктор видаляє об’єкт.

**Lab- 4**

1. Створення базового класу, що в подальшому дозволяє наслідвуватися від нього і забігає копіюванню коду.
2. С++ допускає множинне наслідування, C# ні.
3. Можна тільки задати значення для конструктора батьківського класу.
4. Можна
5. Описання дочернього класу на основі батьківського.
6. Так і дуже часто є обов’язковим.
7. base.methodName();
8. можна.
9. ???
10. Class Figure; Class Romb : Figure;

**Lab- 5**

1. Створення базового класу, що в подальшому дозволяє наслідвуватися від нього і забігає копіюванню коду.
2. С++ допускає множинне наслідування, C# ні.
3. Механізм який дозволяє перевизначення функцій або ж різноманітна реалізація однотипних дій.
4. Метод класу який можна перевизначити в дочірніх.
5. Механізм перевизначення.
6. Для перевизначення в дочірніх класах, якщо ж базова не підходить.
7. Механізм статичного поліморфізму не дозволяє змінювати код під час роботи програми.
8. Динамічний навпаки дозволяє змінювати код під час роботи програми.

**Lab- 6**

1. Клас в якому описують методи та властивості які будуть перевантажені в дочерніх класах .
2. В С++ можна, а в C3# ні.
3. Інтерфейс – це клас в якому створюються прототипи методів, які не можна реалізувати в інтерфейсі .
4. В абстрактному класі можна реалізовувати методи.
5. Коли потрібно розписати поведінку класу написавши методи і їх типи