Github: https://github.com/srijahn/STT_A3

CS 202 Software Tools and Techniques for CSE
Lab 9: Module Dependency and Cohesion Analysis

Overview: This lab focuses on analyzing software dependencies and cohesion in Python and Java projects using pydeps and LCOM. Also understand dependencies and cohesion that help in identifying design flaws and improving software maintainability.

Objectives:

- Use pydeps to generate and analyze dependency graphs for Python projects.
- Use LCOM to measure class cohesion in Java projects.
- Identify design flaws and code smells (anti-patterns).
- Implement modularization and refactoring strategies to optimize software structure.

Environment Setup

- Operating System: Windows/Linux/MacOS
- Programming Languages:
  - Python
  - Java
- Required Tools:
  - pydeps: Used for generating dependency graphs in Python.
  - LCOM: Used to measure class cohesion in Java projects.

Tools and Versions Used

- Python: Version (3.8.10)
- Java: Version (17.0.14)
- Pydeps: v3.0.1
- Apache maven (3.6.3)
- LCOM.jar: For measuring Java class cohesion

Step-by-step procedure, code snippets, outputs, screenshots, and error handling:

1. Install pydeps: **"pip install pydeps"**

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ pip install pydeps
Defaulting to user installation because normal site-packages is not writeable
Collecting pydeps
  Downloading pydeps-3.0.1-py3-none-any.whl.metadata (22 kB)
Collecting stdlib_list (from pydeps)
  Downloading stdlib_list-0.11.1-py3-none-any.whl.metadata (3.3 kB)
Downloading pydeps-3.0.1-py3-none-any.whl (47 kB)
Downloading stdlib_list-0.11.1-py3-none-any.whl (83 kB)
Installing collected packages: stdlib_list, pydeps
Successfully installed pydeps-3.0.1 stdlib_list-0.11.1
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$
```

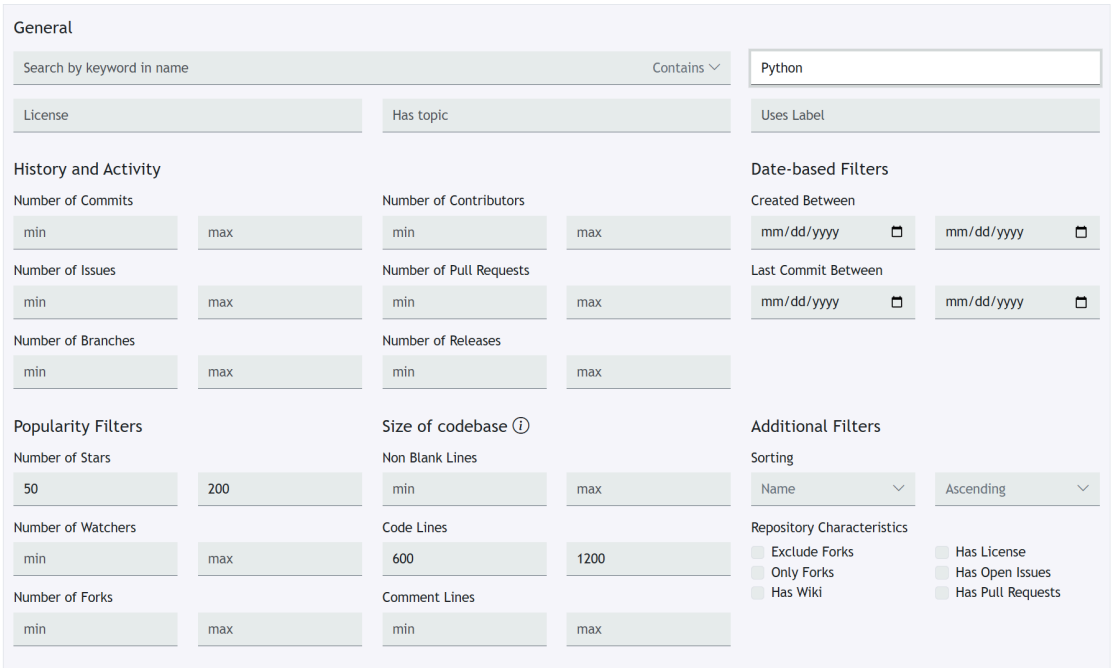
```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ pydeps --version
pydeps v3.0.1
```

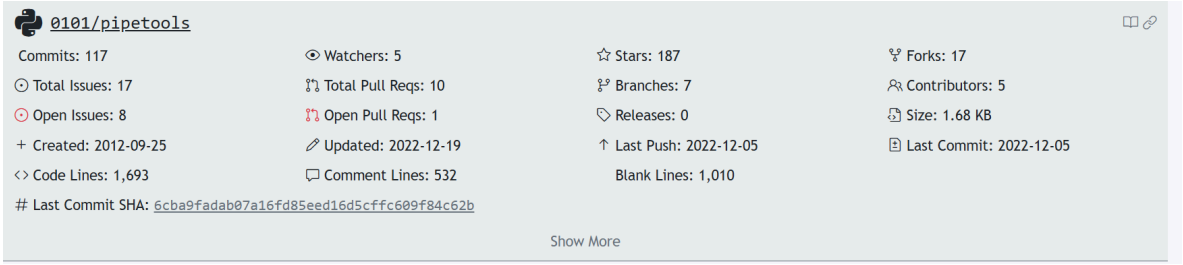
2. Install Java using the command:
   sudo apt update
   sudo apt install openjdk-17-jdk

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-120.04, mixed mode, sharing)
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ python --version
Python 3.8.10
```

3. **Repository selection:**

   Language: Python
   Code lines: 600 - 1200
   Stars: 50 - 200
   I have selected the repository such that it is neither to small for better analysis, neither too large (which makes analysis time and storage consuming), and also such that the repository is popular enough to be a credible one.





4. Cloning the repository:
   Git clone https://github.com/0101/pipetools.git

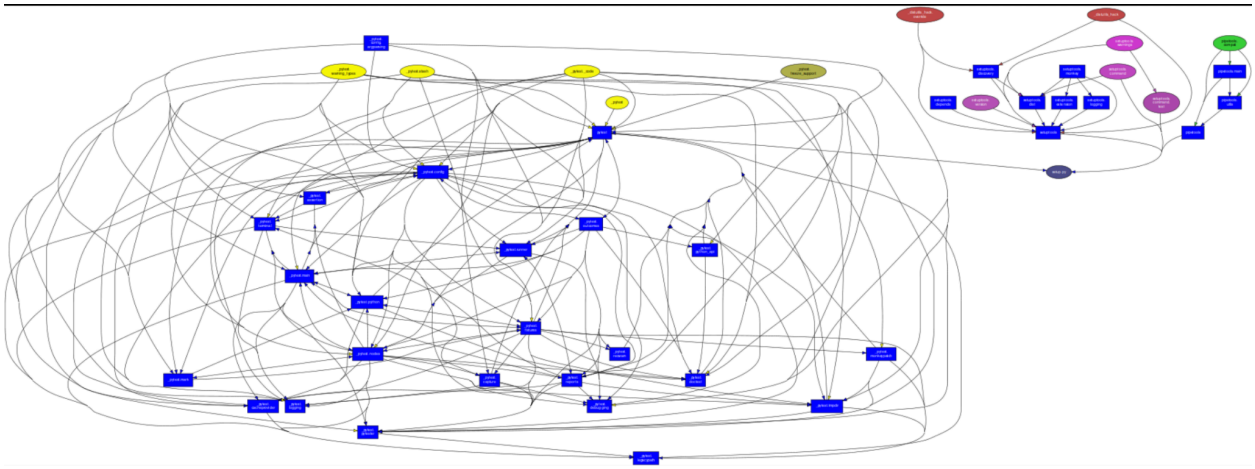
```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ git clone https://github.com/0101/pipetools.git
Cloning into 'pipetools'...
remote: Enumerating objects: 1730, done.
remote: Counting objects: 100% (353/353), done.
remote: Compressing objects: 100% (143/143), done.
remote: Total 1730 (delta 181), reused 324 (delta 170), pack-reused 1377 (from 1)
Receiving objects: 100% (1730/1730), 1.68 MiB | 4.83 MiB/s, done.
Resolving deltas: 100% (1042/1042), done.
```

5. Generate dependency graph:
   Pip install e .
   Pydeps .

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ pip install e .
Defaulting to user installation because normal site-packages is not writeable
Processing /home/set-iitgn-vm/Desktop/lab9/pipetools
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: e in /home/set-iitgn-vm/.local/lib/python3.10/site-packages (1.4.5)
Requirement already satisfied: setuptools>=0.6b1 in /home/set-iitgn-vm/.local/lib/python3.10/site-packages (from pipetools==1.1.0) (76.0.0)
Building wheels for collected packages: pipetools
  Building wheel for pipetools (pyproject.toml) ... done
  Created wheel for pipetools: filename=pipetools-1.1.0-py3-none-any.whl size=13611 sha256=2e04053395ba88695c87ead00d39a9391c0d861a83ee75b9ba15065807a2fe4f
  Stored in directory: /tmp/pip-ephem-wheel-cache-9wssqbpu/wheels/92/cc/e0/3ca1dd17c3f72603d5bbcdf32369edb8852b672298900d5137
Successfully built pipetools
Installing collected packages: pipetools
Successfully installed pipetools-1.1.0
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ pydeps .
dummymodule.py:140: WARNING: SKIPPING ILLEGAL MODULE_NAME: home.set-iitgn-vm.Desktop.lab9.pipetools.setup
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$
```

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ ls
build_scripts  coverage.svg  LICENSE      pipetools   setup.py
changelog.rst  docs          MANIFEST.in  README.rst  test_pipetools
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ pydeps setup.py --show-deps
{
    "_distutils_hack": {
        "bacon": 2,
        "imported_by": [
            "setuptools",
            "setuptools.discovery"
        ],
        "name": "_distutils_hack",
        "path": "/home/set-iitgn-vm/.local/lib/python3.10/site-packages/_distutils_hack/__init__.py"
    },
    "_distutils_hack.override": {
        "bacon": 2,
        "imported_by": [
            "setuptools",
            "setuptools.discovery"
        ],
        "name": "_distutils_hack.override",
        "path": "/home/set-iitgn-vm/.local/lib/python3.10/site-packages/_distutils_hack/override.py"
    },
    "_pytest": {
        "bacon": 2,
        "imported_by": [
            "_pytest.assertion",
```

Pydeps setup.py --show-deps



Store the Dependency Graph JSON: **pydeps setup.py --show-deps > dependencies.txt**
Now, dependencies.txt contains details of module dependencies.
The data is stored in json format:


```
{
    "_distutils_hack": {
        "bacon": 2,
        "imported_by": [
            "setuptools",
            "setuptools.discovery"
        ],
        "name": "_distutils_hack",
        "path": "/home/set-iitgn-vm/.local/l
    },
    "_distutils_hack.override": {
        "bacon": 2,
        "imported_by": [
            "setuptools",
            "setuptools.discovery"
        ],
        "name": "_distutils_hack.override",
        "path": "/home/set-iitgn-vm/.local/l
    },
    "_pytest": {
        "bacon": 2,
        "imported_by": [
            "_pytest.assertion",
            "_pytest.cacheprovider",
            "_pytest.capture",
            "_pytest.config",
            "_pytest.config.argparsing",
            "_pytest.debugging",
            "_pytest.doctest",
            "_pytest.fixtures",
            "_pytest.freeze_support",
            "_pytest.legacypath",
            "_pytest.logging",
            "_pytest.main",
            "_pytest.mark",
            "_pytest.monkeypatch",
```


6. Analysis of the dependency graph:
   - ➔ Identify Highly Coupled Modules:
     - ◆ To identify modules that have too many dependencies, we count:
     - ◆ Fan-out: How many modules a given module depends on.
     - ◆ Fan-in: How many modules depend on a given module.

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ python fan.py

Highly Coupled Modules (Fan-in):
_pytest: 26
_pytest.config: 20
_pytest.nodes: 17
_pytest.config.argparsing: 16
_pytest._code: 13
_pytest.fixtures: 13
_pytest.outcomes: 13
_pytest.main: 11
setuptools: 9
_pytest.reports: 8
_pytest.warning_types: 8
_pytest.stash: 7
_pytest.python: 5
_pytest.terminal: 5
_pytest.assertion: 4
_pytest.capture: 4
_pytest.mark: 4
_pytest.monkeypatch: 4
_pytest.runner: 4
pipetools: 4
pytest: 4
setuptools.monkey: 4
_pytest.cacheprovider: 3
_pytest.tmpdir: 3
pipetools.compat: 3
setuptools.warnings: 3
_distutils_hack: 2
_distutils_hack.override: 2
_pytest.pytester: 2
_pytest.python_api: 2
pipetools.main: 2
setuptools.command: 2
setuptools.discovery: 2
_pytest.debugging: 1
_pytest.doctest: 1
_pytest.freeze_support: 1
_pytest.legacypath: 1
_pytest.logging: 1
_pytest.recwarn: 1
pipetools.utils: 1
setuptools.command.test: 1
setuptools.depends: 1
setuptools.dist: 1
```

◆

```
Modules with Many Dependencies (Fan-out):
pytest: 28
_pytest.pytester: 13
setuptools: 11
_pytest.config: 10
_pytest.doctest: 10
_pytest.fixtures: 10
_pytest.legacypath: 10
_pytest.main: 10
_pytest.python: 10
_pytest.cacheprovider: 9
_pytest.debugging: 9
_pytest.logging: 9
_pytest.nodes: 9
_pytest.runner: 9
_pytest.terminal: 8
_pytest.tmpdir: 8
_pytest.assertion: 6
_pytest.mark: 6
_pytest.reports: 6
_pytest.capture: 5
setup.py: 5
setuptools.dist: 5
pipetools: 4
setuptools.discovery: 3
_pytest.monkeypatch: 3
_pytest.outcomes: 3
_pytest.python_api: 3
_pytest.recwarn: 3
pipetools.utils: 3
_pytest.config.argparsing: 2
pipetools.main: 2
setuptools.command.test: 2
setuptools.extension: 2
setuptools.logging: 2
_pytest.freeze_support: 1
_pytest.warning_types: 1
setuptools.depends: 1
setuptools.monkey: 1
```

```
=== Highly Coupled Modules and Their Dependencies ===
_pytest.fixtures: Fan-In = 13, Fan-Out = 10
  Depends on: ['_pytest', '_pytest._code', '_pytest.config', '_pytest.config.argparsing', '_pytest.main', '_pytest.mark', '_pytest.nodes', '_pytest.outcomes', '_pytest.python',
'_pytest.stash']
  Depended on by: ['_pytest.cacheprovider', '_pytest.capture', '_pytest.doctest', '_pytest.legacypath', '_pytest.logging', '_pytest.main', '_pytest.monkeypatch', '_pytest.nodes',
'_pytest.pytester', '_pytest.python', '_pytest.recwarn', '_pytest.tmpdir', 'pytest']
pytest: Fan-In = 4, Fan-Out = 28
  Depends on: ['_pytest', '_pytest._code', '_pytest.assertion', '_pytest.cacheprovider', '_pytest.capture', '_pytest.config', '_pytest.config.argparsing', '_pytest.debugging',
'_pytest.doctest', '_pytest.fixtures', '_pytest.freeze_support', '_pytest.legacypath', '_pytest.logging', '_pytest.main', '_pytest.mark', '_pytest.monkeypatch', '_pytest.nodes',
'_pytest.outcomes', '_pytest.pytester', '_pytest.python', '_pytest.python_api', '_pytest.recwarn', '_pytest.reports', '_pytest.runner', '_pytest.stash', '_pytest.terminal', '_pytest.tmpdir',
'_pytest.warning_types']
  Depended on by: ['_pytest.config', '_pytest.mark', '_pytest.outcomes', 'setup.py']
_pytest.warning_types: Fan-In = 8, Fan-Out = 1
  Depends on: ['_pytest']
  Depended on by: ['_pytest.cacheprovider', '_pytest.config', '_pytest.doctest', '_pytest.monkeypatch', '_pytest.nodes', '_pytest.pytester', '_pytest.python', 'pytest']
_pytest._code: Fan-In = 13, Fan-Out = 0
  Depends on: []
  Depended on by: ['_pytest.config', '_pytest.debugging', '_pytest.doctest', '_pytest.fixtures', '_pytest.main', '_pytest.nodes', '_pytest.pytester', '_pytest.python', '_pytest.python_api',
'_pytest.reports', '_pytest.runner', '_pytest.terminal', 'pytest']
pytest.cacheprovider: Fan-In = 3, Fan-Out = 9
```

- Modules with High Fan in: These are **core modules** used widely.
- Modules with High Fan out: These modules depend on many others.

➔ Detect cyclic dependencies:

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ nano cyclic.py
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ python cyclic.py

🔄 Detecting Cyclic Dependencies...
Cycle detected: _distutils_hack -> setuptools.discovery -> setuptools.dist -> setuptools -> setuptools.depends -> setuptools
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$
```

We have detected one cycle.

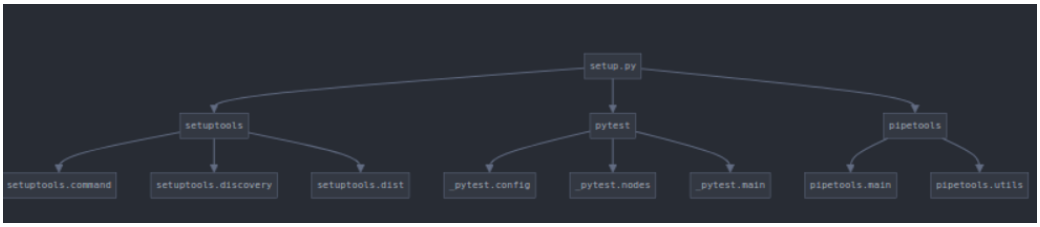
➔ Detect unused modules: setup.py module is isolated.

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ python ana.py

🔄 Detecting Cyclic Dependencies...
🔄 Cycle detected: _distutils_hack -> setuptools.discovery -> setu

🛑 Unused/Isolated Modules:
- setup.py (path: setup.py)
```

➔ Assessing the depth of dependencies:

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ python assess.py

🔄 Detecting Cyclic Dependencies...
🔄 Cycle detected: _distutils_hack -> setuptools.discovery -> setupt

🛑 Unused/Isolated Modules:
- setup.py (path: setup.py)

🔥 Highly Coupled Modules (Fan-in count):
- _pytest: 26 modules depend on it.
- _pytest.config: 20 modules depend on it.
- _pytest.nodes: 17 modules depend on it.
- _pytest.config.argparsing: 16 modules depend on it.
- _pytest._code: 13 modules depend on it.
- _pytest.fixtures: 13 modules depend on it.
- _pytest.outcomes: 13 modules depend on it.
- _pytest.main: 11 modules depend on it.
- setuptools: 9 modules depend on it.
- _pytest.reports: 8 modules depend on it.
- _pytest.warning_types: 8 modules depend on it.
```

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/pipetools$ python assess.py
Dependency Depth Analysis Report
==============================

Maximum Overall Dependency Depth: 4

Top 5 Most Deep Modules:
_distutils_hack: Depth = 4
_distutils_hack.override: Depth = 4
_pytest: Depth = 4
_pytest._code: Depth = 4
_pytest.assertion: Depth = 4

Top 5 Most Connected Modules:
pytest: Total Connections = 58 (Incoming: 29, Outgoing: 29)
_pytest: Total Connections = 52 (Incoming: 26, Outgoing: 26)
_pytest.config: Total Connections = 48 (Incoming: 24, Outgoing: 24)
_pytest.nodes: Total Connections = 46 (Incoming: 23, Outgoing: 23)
_pytest.fixtures: Total Connections = 40 (Incoming: 20, Outgoing: 20)
```



➔ Cyclic Dependencies and Maintainability Issues:

Cyclic dependencies occur when two or more modules depend on each other directly or indirectly.

Cycle detected: _distutils_hack -> setuptools.discovery -> setuptools.dist -> setuptools -> setuptools.logging -> setuptools

- _distutils_hack depends on setuptools.discovery
- setuptools.discovery depends on setuptools.dist
- setuptools.dist depends on setuptools
- setuptools depends back on setuptools.dist, completing a cycle.

Problems Caused by Cyclic Dependencies:

1. **Harder to Understand and Debug:** When a module depends on another that eventually depends back on the original, it creates hidden complexity. If we modify one module, it could unexpectedly break another module higher up in the cycle.
2. **Difficult to Refactor or Reuse:** We can't easily extract a module for independent use because it's tightly coupled with others. Breaking one dependency could cascade errors across multiple modules.
3. **Risk of Infinite Loops or Import Errors:** If Python tries to import modules that depend on each other in a cycle, it may fail to resolve imports or get stuck in a loop.
4. **Testing Becomes More Fragile:** Unit testing individual modules becomes difficult because testing one may require mocking several dependent modules, increasing complexity.

5. **Build and Deployment Issues:** Dependency cycles can cause delays in dependency resolution, making it harder to deploy updates.

Fan-in and Fan-out:

- From my analysis: _pytest has a fan-in of 26, so 26 other modules depend on it.
- Impact:
    - If _pytest changes, 26 modules might break.
    - Makes refactoring very risky.
    - Should consider breaking it into smaller, loosely coupled submodules.
- If a module has high fan-out, it means it relies on too many other modules.
- Impact:
    - If one dependency fails or changes, the module may stop working.
    - Such modules become fragile and hard to maintain.

How to Improve Maintainability:

1. Break Cycles Using Abstraction
    - Introduce interface layers or event-driven mechanisms to decouple modules.
    - Example: Instead of _distutils_hack directly using setuptools.dist, introduce an intermediate API.
2. Reduce Fan-in with Modularization
    - Large core modules (*e.g., _pytest*) can be split into submodules to distribute dependencies.
3. Reduce Fan-out by Limiting Direct Dependencies
    - Instead of one module importing everything, use dependency injection to control how dependencies are used.

➔ How would changes in the core module affect the rest of the system?

Identifying Core Modules: Core modules are those with high fan-in values:

- **_pytest** (26 dependencies)
- **_pytest.config** (20 dependencies)
- **_pytest.nodes** (17 dependencies)
- **_pytest.config.argparsing** (16 dependencies)
- **_pytest._code** (13 dependencies)

Impact of Changing Core Modules: If these modules change, the risk of system-wide failure is high because they are used across many components. Any modifications in these modules should be carefully tested before deployment.

- Modifying _pytest may break all modules that depend on it.
- Refactoring _pytest.config could disrupt argument parsing.
- A breaking change in _pytest.nodes could cause test node management failures.

Identifying High-Risk Modules: Modules with both high fan-in and high fan-out are most fragile. They introduce unintended side effects in deeply dependent modules and Cascade failures across multiple components when modified.

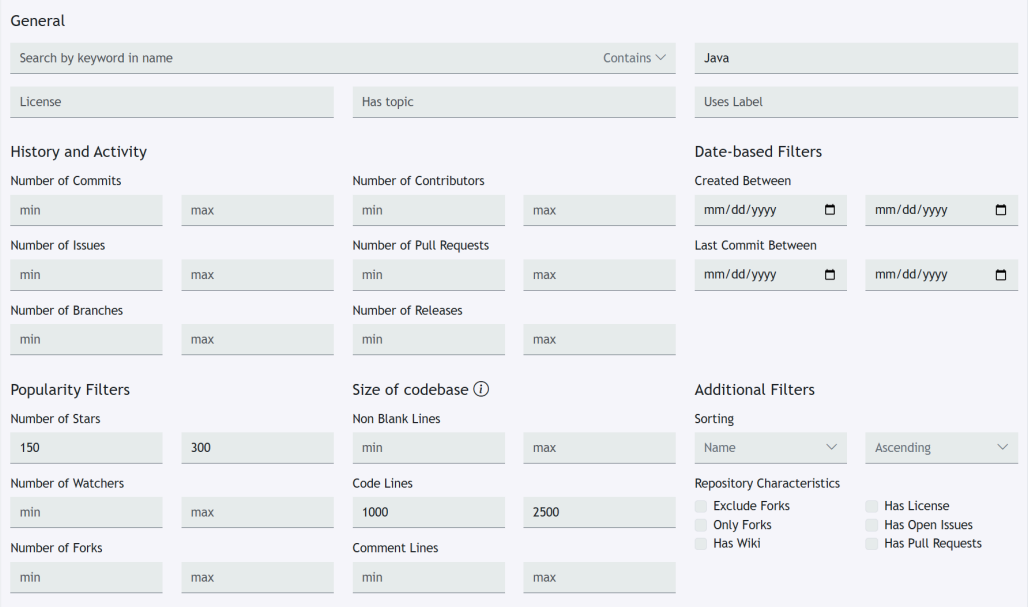
From my data, high fan-out modules (many dependencies) include:

- **setuptools** (depends on multiple other modules).
- **_pytest.config** (modifying it could affect many settings).
- **_pytest.fixtures** (affects dependency injection in tests).

Risk if These Modules Change:

- Changing **setuptools** might break the entire package management system.
- Modifying **_pytest.config** could cause test framework failures.
- Altering **_pytest.fixtures** could disrupt unit testing behavior.

7. Java Repository selection:
   Language: Java
   Code lines: 1000 - 2500
   Stars: 150 - 300
   I have selected the repository such that it is neither to small for better analysis, neither too large (which makes analysis time and storage consuming), and also such that the repository is popular enough to be a credible one.



Selected repository: https://github.com/7heaven/UILibrary.git

8. Cloning the repository: **git clone https://github.com/7heaven/UILibrary.git**

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ git clone https://github.com/7heaven/UILibrary.git
Cloning into 'UILibrary'...
remote: Enumerating objects: 489, done.
remote: Total 489 (delta 0), reused 0 (delta 0), pack-reused 489 (from 1)
Receiving objects: 100% (489/489), 7.92 MiB | 1.68 MiB/s, done.
Resolving deltas: 100% (135/135), done.
```

Checking for the availability of JAVA and downloading LCOM.jar and extract it or even cloning the LCOM from git clone https://github.com/tushartushar/LCOM.git and create the LCOM.jar

file:

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ git clone https://github.com/tushartus
har/LCOM.git
Cloning into 'LCOM'...
remote: Enumerating objects: 2064, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 2064 (delta 2), reused 16 (delta 1), pack-reused 2043 (from 1)
Receiving objects: 100% (2064/2064), 2.38 MiB | 2.67 MiB/s, done.
Resolving deltas: 100% (1123/1123), done.
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ cd LCOM
bash: cd: LCOM: No such file or directory
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ ls
pipetools  UILibrary
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ cd LCOM
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/LCOM$ ls
lib  LICENSE  manifest.mf  pom.xml  README.md  src  tests
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/LCOM$ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 1.8.0_442, vendor: Private Build, runtime: /usr/lib/jvm/java-8-ope
```

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/LCOM$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< LCOM:LCOM >----------------------------
[INFO] Building LCOM 1.0.0
[INFO] --------------------------------[ jar ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/
plugins/maven-resources-plugin/2.6/maven-resources-plugin-2.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/p
lugins/maven-resources-plugin/2.6/maven-resources-plugin-2.6.pom (8.1 kB at 3.5
kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/
```

```
with assembly file: /home/set-iitgn-vm/Desktop/lab9/LCOM/target/LCOM.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  49.294 s
[INFO] Finished at: 2025-03-25T05:06:50+05:30
[INFO] ------------------------------------------------------------------------
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/LCOM$ ls target/
archive-tmp  generated-sources  LCOM.jar        maven-status
classes      LCOM-1.0.0.jar     maven-archiver  test-classes
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/LCOM$
```

Checking if the requirements are present:

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/UILibrary$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-120.04, mixed mode, sharing)
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/UILibrary$ cd UILibrary/
```

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/UILibrary$ javac -version
javac 17.0.14
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/UILibrary$
```

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ sudo apt install default-jdk
[sudo] password for set-iitgn-vm:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required
  chromium-codecs-ffmpeg-extra gir1.2-goa-1.0 gstreamer1.0-vaapi libgstreamer-
  linux-image-5.15.0-130-generic linux-modules-5.15.0-130-generic linux-module
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  default-jdk-headless default-jre default-jre-headless openjdk-11-jdk openjdk
Suggested packages:
  openjdk-11-demo openjdk-11-source visualvm fonts-ipafont-gothic fonts-ipafon
The following NEW packages will be installed:
  default-jdk default-jdk-headless default-jre default-jre-headless openjdk-11
0 upgraded, 8 newly installed, 0 to remove and 54 not upgraded.
```

9. **Download LCOM.jar.**
10. LCOM analysis:
    Command: java -jar LCOM.jar -i UILibrary/uilibrary/src/main/java -o lcom_results
    - LCOM/target/LCOM.jar → This is the LCOM analysis tool (JAR file) that you're executing.
      Input Path (-i UILibrary/uilibrary/src/main/java):
    - UILibrary/uilibrary/src/main/java → This is the source code directory that contains Java files you want to analyze.
    - It contains Java classes for which LCOM (Lack of Cohesion of Methods) metrics will be calculated. Output Path (-o lcom_results):
    - lcom_results → This is the folder where the LCOM results will be stored.
    - The tool will generate CSV files (or other formats) with the LCOM metrics inside this directory.

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ mv LCOM/target/LCOM.jar .
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ java -jar LCOM.jar -i UILibrary/uilibr
ary/src/main/java -o lcom_results
Parsing the source code ...
Resolving symbols...
Computing metrics...
Done.
```

The results are in a folder named lcom_results.

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9$ ls lcom_results/
LcomLog25032025_0530.txt  TypeMetrics.csv
```

The csv file:

	A	B	C	D	E	F	G	H	I	J
1	Project Name	Package Name	Type Name	LCOM1	LCOM2	LCOM3	LCOM4	LCOM5	YALCOM	
2	java	com.sevenheaven.uilibrary.drawables	GroupedAvatarDrawable	39	23	2	2	0.814285714285714	0.181818181818182	
3	java	com.sevenheaven.uilibrary.drawables	RoundEdgeDrawable	18	8	4	3	0.885714285714286	0.375	
4	java	com.sevenheaven.uilibrary.drawables	MaskDrawable	30	5	3	2	0.74	0.181818181818182	
5	java	com.sevenheaven.uilibrary.drawables.progressive.providers	PathProgressProvider	41	27	5	3	0.872222222222222	0.272727272727273	
6	java	com.sevenheaven.uilibrary.drawables.progressive.providers	PathDesc	2	1	3	3	0.875	0.666666666666667	
7	java	com.sevenheaven.uilibrary.drawables.progressive.providers	AppStoreStyleProgressProvider	3	0	1	1	0.704545454545455	0	
8	java	com.sevenheaven.uilibrary.drawables.progressive	LoadingDrawable	78	78	14	13	1.02777777777778	0.846153846153846	
9	java	com.sevenheaven.uilibrary.drawables.progressive	ProgressiveDrawable	258	216	11	11	0.919270833333333	0.44	
10	java	com.sevenheaven.uilibrary.drawables.progressive	DrawContentProvider	9	8	4	4	0.75	0.8	
11	java	com.sevenheaven.uilibrary.shapes	ArcShape	20	12	3	1	0.75	0	
12	java	com.sevenheaven.uilibrary.shapes	PolygonShape	41	27	4	3	0.828571428571429	0.272727272727273	
13	java	com.sevenheaven.uilibrary.utils	GeomUtil	10	0	5	3	0	0.6	
14	java	com.sevenheaven.uilibrary.utils	PathMeasurement	10	0	2	1	0.666666666666667	0	
15	java	com.sevenheaven.uilibrary.views	AnimatedImageView	32	19	4	3	0.777777777777778	0.3	
16	java	com.sevenheaven.uilibrary.views	PageIndicator	99	78	4	3	0.908333333333333	0.1875	
17	java	com.sevenheaven.uilibrary.views	GroupedAvatarImageView	15	15	6	6	1	1	
18	java	com.sevenheaven.uilibrary.views	FitWidthImageView	83	61	7	4	0.814285714285714	0.266666666666667	
19										

There are 16 different classes.

11. **Identification of classes with highest LCOM values:**

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/lcom_results$ python ana.py
Column names after fixing: Index(['Project Name', 'Package Name', 'Type Name', 'LCOM1', 'LCOM2', 'LCOM3',
       'LCOM4', 'LCOM5', 'YALCOM'],
      dtype='object')

High LCOM Classes Identified:
   Project Name  ...    YALCOM
6          java  ...  0.846154
7          java  ...  0.440000
14         java  ...  0.187500
16         java  ...  0.266667

[4 rows x 9 columns]

Cohesion table saved to HighLCOMClasses.csv
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/lcom_results$
```

CSV file:

	A	B	C	D	E	F	G	H	I
1	Project Name	Package Name	Type Name	LCOM1	LCOM2	LCOM3	LCOM4	LCOM5	YALCOM
2	java	com.sevenheaven.uilibrary.drawables.progressive	LoadingDrawable	78	78	14	13	1.02777777777778	0.846153846153846
3	java	com.sevenheaven.uilibrary.drawables.progressive	ProgressiveDrawable	258	216	11	11	0.919270833333333	0.44
4	java	com.sevenheaven.uilibrary.views	PageIndicator	99	78	4	3	0.908333333333333	0.1875
5	java	com.sevenheaven.uilibrary.views	FitWidthImageView	83	61	7	4	0.814285714285714	0.266666666666667
6									

Code for the same:

```python
import pandas as pd

# Load the CSV file with the correct separator
csv_file = "TypeMetrics.csv"
df = pd.read_csv(csv_file, sep=",")

# Debugging: Print column names again
print("Column names after fixing:", df.columns)

# Strip any leading/trailing spaces from column names
df.columns = df.columns.str.strip()

# Define thresholds for "High LCOM"
LCOM1_THRESHOLD = 50
LCOM2_THRESHOLD = 40

# Ensure columns exist before filtering
required_columns = {"LCOM1", "LCOM2"}
if not required_columns.issubset(df.columns):
    print(f"Error: Missing required columns. Available columns: {df.columns}")
else:
    # Filter classes with high LCOM values
    high_lcom_classes = df[(df["LCOM1"] > LCOM1_THRESHOLD) | (df["LCOM2"] > LCOM2_THRESHOLD)]

    # Create a cohesion table with selected columns
    cohesion_table = high_lcom_classes[["Project Name", "Package Name", "Type Name", "LCOM1", "LCOM2", "LCOM3", "LCOM4", "LCOM5", "YALCOM"]]

    # Save the table to a new CSV file
    output_file = "HighLCOMClasses.csv"
    cohesion_table.to_csv(output_file, index=False)

    # Print the results
    print("\nHigh LCOM Classes Identified:")
    print(cohesion_table)
    print(f"\nCohesion table saved to {output_file}")
```

My thresholds are 50 for LCOM1 and 40 for LCOM2

12. **Analysis:**

What does a high LCOM value suggest?

A high LCOM (Lack of Cohesion of Methods) value indicates that:

- The class has low cohesion, meaning its methods do not share many common instance variables.
- The class is likely too broad and should be split into smaller, more focused classes.
- It may be harder to maintain, test, and reuse due to unrelated functionalities being grouped together.

Is there a chance for performing functional decomposition?

Yes, classes with high LCOM values should be analyzed for possible functional decomposition:

- If a class has many independent methods that don't interact with each other, it can be split into multiple smaller classes, each handling a specific responsibility.
- This improves modularity, readability, and reusability of the code.

13. **Creating cohesion table:**

```
set-iitgn-vm@set-iitgn-vm:~/Desktop/lab9/lcom_results$ python cohe.py

 **Cohesion Analysis Table**

                      Type Name  LCOM1  LCOM2  LCOM3  LCOM4    LCOM5    YALCOM
          GroupedAvatarDrawable   39.0   23.0    2.0    2.0 0.814286 0.181818
              RoundEdgeDrawable   18.0    8.0    4.0    3.0 0.885714 0.375000
                   MaskDrawable   30.0    5.0    3.0    2.0 0.740000 0.181818
           PathProgressProvider   41.0   27.0    5.0    3.0 0.872222 0.272727
                       PathDesc    2.0    1.0    3.0    3.0 0.875000 0.666667
AppStoreStyleProgressProvider      3.0    0.0    1.0    1.0 0.704545 0.000000
                LoadingDrawable   78.0   78.0   14.0   13.0 1.027778 0.846154
            ProgressiveDrawable  258.0  216.0   11.0   11.0 0.919271 0.440000
            DrawContentProvider    9.0    8.0    4.0    4.0 0.750000 0.800000
                       ArcShape   20.0   12.0    3.0    1.0 0.750000 0.000000
                   PolygonShape   41.0   27.0    4.0    3.0 0.828571 0.272727
                        GeomUtil   10.0    0.0    5.0    3.0 0.000000 0.600000
                PathMeasurement   10.0    0.0    2.0    1.0 0.666667 0.000000
              AnimatedImageView   32.0   19.0    4.0    3.0 0.777778 0.300000
                  PageIndicator   99.0   78.0    4.0    3.0 0.908333 0.187500
          GroupedAvatarImageView   15.0   15.0    6.0    6.0 1.000000 1.000000
              FitWidthImageView   83.0   61.0    7.0    4.0 0.814286 0.266667
```

	A	B	C	D	E	F	G	H	I
1	Project Name	Package Name	Type Name	LCOM1	LCOM2	LCOM3	LCOM4	LCOM5	YALCOM
2	java	com.sevenheaven.uilibrary.drawables	GroupedAvatarDrawable	39	23	2	2	0.814285714285714	0.181818181818182
3	java	com.sevenheaven.uilibrary.drawables	RoundEdgeDrawable	18	8	4	3	0.885714285714286	0.375
4	java	com.sevenheaven.uilibrary.drawables	MaskDrawable	30	5	3	2	0.74	0.181818181818182
5	java	com.sevenheaven.uilibrary.drawables.progressive.providers	PathProgressProvider	41	27	5	3	0.872222222222222	0.272727272727273
6	java	com.sevenheaven.uilibrary.drawables.progressive.providers	PathDesc	2	1	3	3	0.875	0.666666666666667
7	java	com.sevenheaven.uilibrary.drawables.progressive.providers	AppStoreStyleProgressProvider	3	0	1	1	0.704545454545455	0
8	java	com.sevenheaven.uilibrary.drawables.progressive	LoadingDrawable	78	78	14	13	1.02777777777778	0.846153846153846
9	java	com.sevenheaven.uilibrary.drawables.progressive	ProgressiveDrawable	258	216	11	11	0.919270833333333	0.44
10	java	com.sevenheaven.uilibrary.drawables.progressive	DrawContentProvider	9	8	4	4	0.75	0.8
11	java	com.sevenheaven.uilibrary.shapes	ArcShape	20	12	3	1	0.75	0
12	java	com.sevenheaven.uilibrary.shapes	PolygonShape	41	27	4	3	0.828571428571429	0.272727272727273
13	java	com.sevenheaven.uilibrary.utils	GeomUtil	10	0	5	3	0	0.6
14	java	com.sevenheaven.uilibrary.utils	PathMeasurement	10	0	2	1	0.666666666666667	0
15	java	com.sevenheaven.uilibrary.views	AnimatedImageView	32	19	4	3	0.777777777777778	0.3
16	java	com.sevenheaven.uilibrary.views	PageIndicator	99	78	4	3	0.908333333333333	0.1875
17	java	com.sevenheaven.uilibrary.views	GroupedAvatarImageView	15	15	6	6	1	1
18	java	com.sevenheaven.uilibrary.views	FitWidthImageView	83	61	7	4	0.814285714285714	0.266666666666667
19									

Code for the above:

```python
import pandas as pd

csv_file = "TypeMetrics.csv"
df = pd.read_csv(csv_file, sep=",")

df.columns = df.columns.str.strip()

# Ensure required columns exist
required_columns = {"Project Name", "Package Name", "Type Name", "LCOM1", "LCOM2", "LCOM3", "LCOM4", "LCOM5", "YALCOM"}
if not required_columns.issubset(df.columns):
    print(f"Error: Missing required columns. Available columns: {df.columns}")
else:
    # Select required columns
    cohesion_data = df[
        ["Type Name", "LCOM1", "LCOM2", "LCOM3", "LCOM4", "LCOM5", "YALCOM"]
    ]

    # Print table format
    print("\n **Cohesion Analysis Table** \n")
    print(cohesion_data.to_string(index=False))
```

Discussion and Conclusion:

Challenges Faced:

- Understanding and interpreting cyclic dependencies.
- Configuring pydeps for complex project structures.
- Searching a repository with atleast 10 classes.
- Navigating the source and input file for LCOM analysis.
- Interpreting different LCOM values and their implications.

Reflections and Lessons Learned:

- High module coupling in Python projects increases maintenance challenges.
- Cohesion analysis helps in refactoring Java classes for better modularization.
- Dependency graphs visually aid in identifying potential refactoring areas.

Summary: This lab provided hands-on experience in analyzing module dependencies and class cohesion. By using tools like pydeps and LCOM, I was able to effectively identify design flaws, assess dependency impact, and suggest improvements. The insights gained from this lab can be applied to optimize software design and enhance maintainability.

Resources:

- https://drive.google.com/file/d/12wlXeIC5bMd2mPCNyAp2UU1T_RFTfSUC/view
- https://pypi.org/project/modulegraph/
- https://github.com/thebjorn/pydeps
- https://github.com/tushartushar/LCOM

CS202 Software Tools and Techniques for CSE
Lab 10: Development of C# Console Applications

Overview and Objectives

The primary objectives are:

- Set up and using Visual Studio for .NET development.
- Write and execute basic C# console applications.
- Implement fundamental programming constructs: loops, conditionals, and functions.
- Apply object-oriented programming principles in C#.
- Understand the benefits of the Visual Studio Debugger.

Environment Setup, Tools and Versions:

- Operating System: Windows 11
- Software: Visual Studio Code 2022 (Community Edition) with .NET SDK
- Programming Language: C# ( )

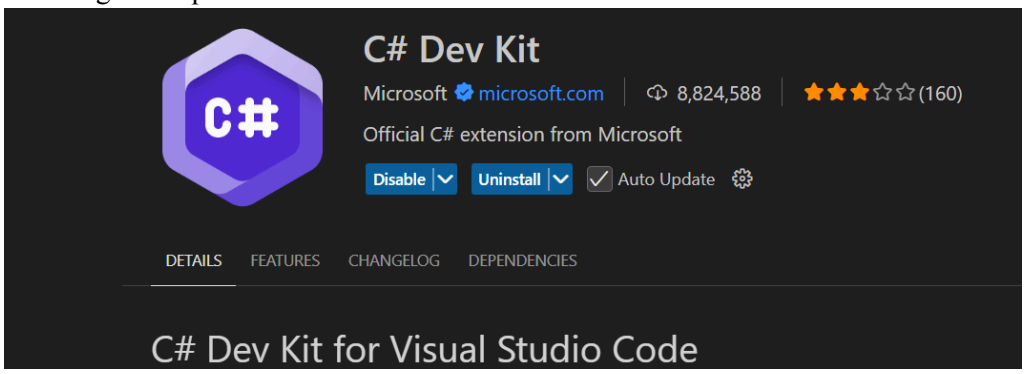
Discussion and Conclusion:

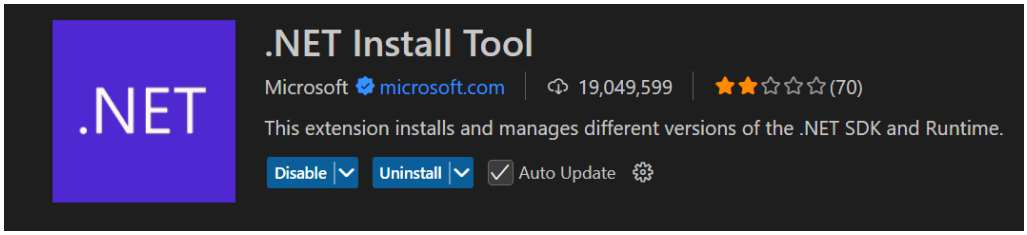
Challenges Faced:

- Initial setup of Visual Studio and ensuring the correct .NET SDK version.
- Handling user input validation and exceptions effectively.
- Understanding object-oriented concepts in C# and implementing inheritance correctly.
- Debugging logic errors and runtime exceptions.

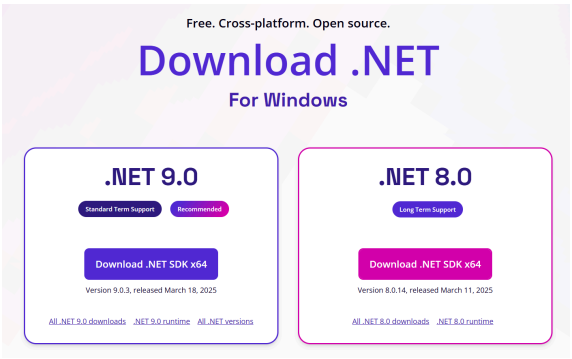
Step by Step procedure, code snippets, outputs, screenshots, and error handling:

1. Installing the required extensions in vs code:





Downloading and installing .NET 9.0 from the link: https://dotnet.microsoft.com/en-us/download



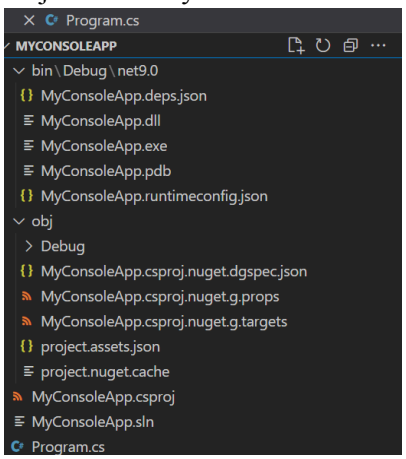

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9> dotnet --version
9.0.202
```

2. **STEP 1**: Creating a new project and navigating to it now:

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9> dotnet new console -n MyConsoleApp
```

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9> cd  MyConsoleApp
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp>
```

Project directory:



Program.cs:

```
Program.cs
1   // See https://aka.ms/new-console-template for more information
2   Console.WriteLine("Hello, World!");
3
```

Running this simple code in Program.cs:

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run
Hello, World!
```

3. **Step 2: Understanding Basic Syntax and Control Structures.**

BasicOp.cs > BasicOperations > Main

```
using System;
0 references
class BasicOperations
{
    0 references
    static void Main()
    {
        Console.Write("Enter first number: ");   // accept user input
        int num1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter second number: ");
        int num2 = Convert.ToInt32(Console.ReadLine());

        // Perform operations
        int sum = num1 + num2;
        int difference = num1 - num2;
        int product = num1 * num2;
        double quotient = (num2 != 0) ? (double)num1 / num2 : double.NaN;

        string sumType = (sum % 2 == 0) ? "even" : "odd"; // Check if sum is even or odd

        // Results
        Console.WriteLine($"Addition: {sum} ({sumType})");
        Console.WriteLine($"Subtraction: {difference}");
        Console.WriteLine($"Multiplication: {product}");
        Console.WriteLine($"Division: {(num2 != 0 ? quotient.ToString() : "Cannot divide by zero")}");
    }
}
```

Building and running the code:

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet build
Restore complete (1.3s)
  MyConsoleApp succeeded (0.7s) → bin\Debug\net9.0\MyConsoleApp.dll

Build succeeded in 2.5s
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run
Running Basic Operations...
Enter first number: 6
Enter second number: 9
Addition: 15 (odd)
Subtraction: -3
Multiplication: 54
Division: 0.6666666666666666
```

4. **Step 3: Loops and Functions**

Program.cs | BasicOp.cs | loopFunc.cs

loopFunc.cs > LoopsFunctions > Run

```
using System;

1 reference
class LoopsFunctions
{
    1 reference
    public static void Run()
    {
        Console.WriteLine("Numbers from 1 to 10:");
        for (int i = 1; i <= 10; i++)
        {
            Console.Write(i + " ");
        }
        Console.WriteLine("\n");

        string input;
        do
        {
            Console.Write("Enter something (type 'exit' to quit): ");
            input = Console.ReadLine() ?? string.Empty;
        } while (input.ToLower() != "exit");

        Console.Write("Enter a number to calculate factorial: ");
        int number = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine($"Factorial of {number} is {Factorial(number)}");
    }

    1 reference
    private static long Factorial(int n)
    {
        long fact = 1;
        for (int i = 1; i <= n; i++)
        {
            fact *= i;
        }
        return fact;
    }
}
```

Building and Running:

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet build
Restore complete (1.3s)
  MyConsoleApp succeeded (0.9s) → bin\Debug\net9.0\MyConsoleApp.dll

Build succeeded in 3.0s
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run
Running Basic Operations...
Numbers from 1 to 10:
1 2 3 4 5 6 7 8 9 10

Enter something (type 'exit' to quit): exit
Enter a number to calculate factorial: 17
Factorial of 17 is 355687428096000
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp>
```

5. Step 4: Object-Oriented Programming:

Student.cs > Student

```
using System;

1 reference
class Student
{
    2 references
    public string Name { get; set; }
    2 references
    public int ID { get; set; }
    6 references
    public double Marks { get; set; }

    0 references
    public Student(string name, int id, double marks)
    {
        Name = name;
        ID = id;
        Marks = marks;
    }

    1 reference
    public string GetGrade()
    {
        if (Marks >= 90) return "A";
        else if (Marks >= 80) return "B";
        else if (Marks >= 70) return "C";
        else if (Marks >= 60) return "D";
        else return "F";
    }

    0 references
    public void DisplayDetails()
    {
        Console.WriteLine($"Student Name: {Name}, ID: {ID}, Marks: {Marks}, Grade: {GetGrade()}");
    }
}
```

iitgn.cs > ...

```
1   using System;
2
    3 references
3   class StudentIITGN : Student
4   {
        2 references
5       public string Hostel_Name_IITGN { get; set; }
6
        1 reference
7       public StudentIITGN(string name, int id, double marks, string hostel)
8           : base(name, id, marks)
9       {
10          Hostel_Name_IITGN = hostel;
11      }
12
        1 reference
13      public void DisplayIITGNDetails()
14      {
15          Console.WriteLine($"IITGN Student: {Name}, ID: {ID}, Marks: {Marks}, Grade: {GetGrade()}, Hostel: {Hostel_Name_I
16      }
17  }
18
```

Building and Running:

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet build
Restore complete (1.2s)
  MyConsoleApp succeeded (0.9s) → bin\Debug\net9.0\MyConsoleApp.dll

Build succeeded in 2.7s
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run
Student Name: John Doe, ID: 12345, Marks: 85.5, Grade: B
```

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet build
Restore complete (1.2s)
  MyConsoleApp succeeded (0.8s) → bin\Debug\net9.0\MyConsoleApp.dll

Build succeeded in 2.6s
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run
IITGN Student: Jane Doe, ID: 67890, Marks: 92, Grade: A, Hostel: Hostel A
```

6. **Step 5: Exception Handling (`ExceptionHandling.cs`):**

exception.cs > ExceptionHandling > Run

```
using System;
0 references
class ExceptionHandling
{
    0 references
    public static void Run()
    {
        try
        {
            Console.Write("Enter first number: ");
            int num1 = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter second number: ");
            int num2 = Convert.ToInt32(Console.ReadLine());

            int sum = num1 + num2;
            Console.WriteLine($"Sum: {sum} ({(sum % 2 == 0 ? "Even" : "Odd")})");
            Console.WriteLine($"Multiplication: {num1 * num2}");
            Console.WriteLine(num2 != 0 ? $"Division: {num1 / num2}" : "Cannot divide by zero");
        }
        catch (FormatException)
        {
            Console.WriteLine("Invalid input! Please enter a number.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}
```

Building and Running code:

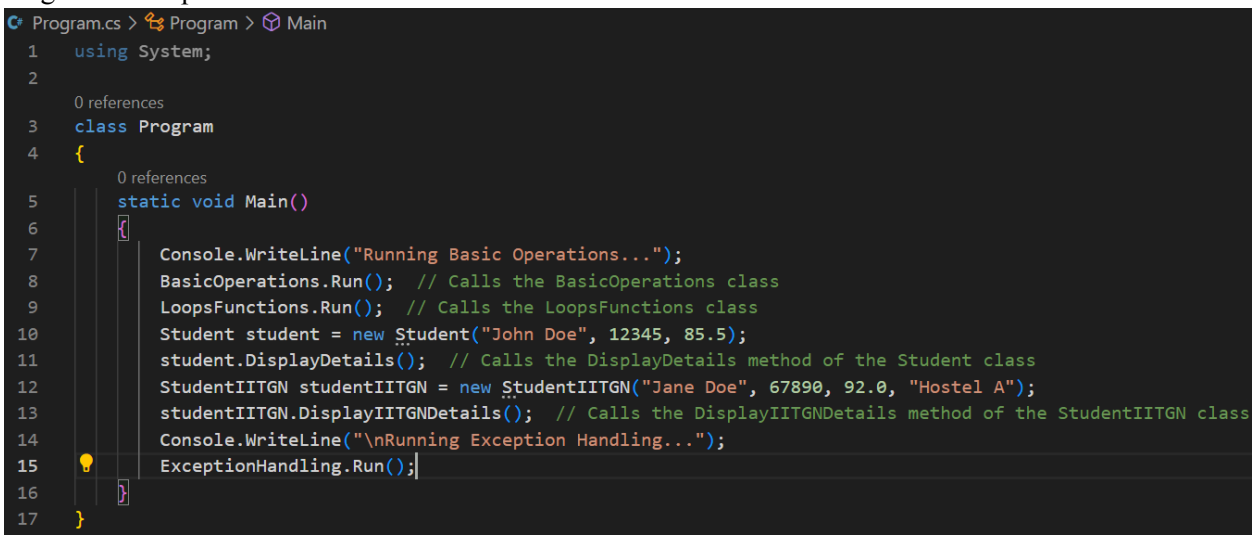
```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet build
Restore complete (1.2s)
Restore complete (1.2s)
  MyConsoleApp succeeded (0.8s) → bin\Debug\net9.0\MyConsoleApp.dll

Build succeeded in 2.6s
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run

Running Exception Handling...
Enter first number: 6
Enter second number: 3
Sum: 9 (Odd)
Multiplication: 18
Division: 2
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run

Running Exception Handling...
Enter first number: 12
Enter second number: 0
Sum: 12 (Even)
Multiplication: 0
Cannot divide by zero
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run

Running Exception Handling...
Enter first number: 0
Enter second number: 4
Sum: 4 (Even)
Multiplication: 0
Division: 0
```

```
PS C:\Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp> dotnet run

Running Exception Handling...
Enter first number: 8
Enter second number: Hii
Invalid input! Please enter a number.
```

Two exceptions are being handled:

- When denominator is zero, it is showing that we can not divide by zero.
- When we give invalid input, like a string instead of an integer for operation, the program is not crashing and it is telling that it's an invalid input.

7. Program.cs is updated so that all these new files can be run:



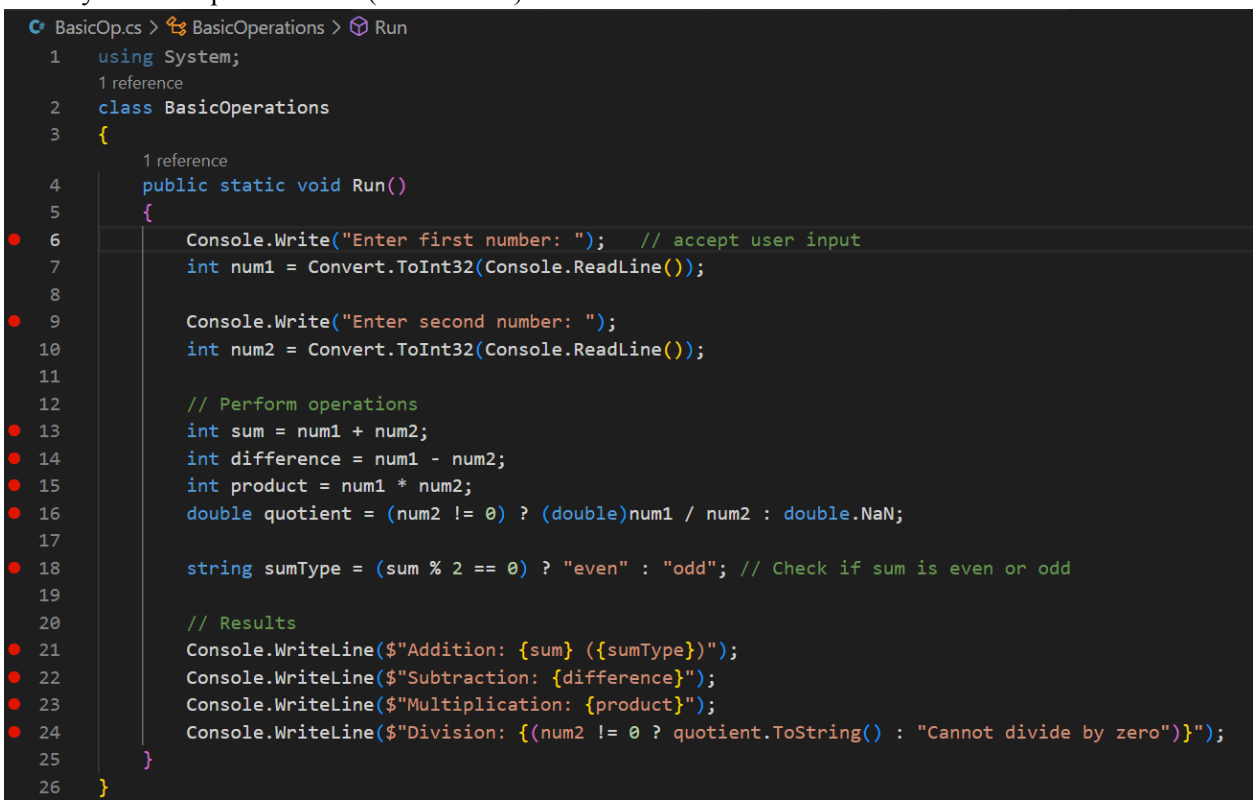
```
Program.cs > Program > Main
1   using System;
2
    0 references
3   class Program
4   {
        0 references
5       static void Main()
6       {
7           Console.WriteLine("Running Basic Operations...");
8           BasicOperations.Run();  // Calls the BasicOperations class
9           LoopsFunctions.Run();  // Calls the LoopsFunctions class
10          Student student = new Student("John Doe", 12345, 85.5);
11          student.DisplayDetails();  // Calls the DisplayDetails method of the Student class
12          StudentIITGN studentIITGN = new StudentIITGN("Jane Doe", 67890, 92.0, "Hostel A");
13          studentIITGN.DisplayIITGNDetails();  // Calls the DisplayIITGNDetails method of the StudentIITGN class
14          Console.WriteLine("\nRunning Exception Handling...");
15          ExceptionHandling.Run();
16      }
17  }
```

8. **Steps to Debug**
   - ➔ **Open Visual Studio Code**
     - ◆ The C# Console Application is already present as we have configured in the starting.
   - ➔ **Set Breakpoints**
     - ◆ By clicking on the left margin next to a line of code or even by pressing **F9** we can insert a breakpoint.
     - ◆ Breakpoints were placed at the following key operations:
       - Before user input is read.
       - Before arithmetic operations.
       - Before conditionals checking even/odd.
       - Before entering loops and function calls.
   - ➔ **Start Debugging**
     - ◆ Press F5 or go to **Debug > Start Debugging** to launch application in Debug mode.
   - ➔ **Use Debugging Operations**
     - ◆ **Step-In (F11)**: It is used to call functions like Factorial() to analyze their execution.
     - ◆ **Step-Over (F10)**: It is used to execute a line and move to the next without entering functions.
     - ◆ **Step-Out (Shift+F11)**: Used to exit from a function back to the main program.
     - ◆ **Watch Window**: Used to monitor variable values in real-time.
   - ➔ **Inspect Variable Values**
     - ◆ Hover over variables to see their values.
     - ◆ Use the **Locals** and **Autos** windows to monitor variable states.
   - ➔ **Continue or Stop Execution**
     - ◆ Press F5 to continue execution.
     - ◆ Use **Stop Debugging (Shift+F5)** to terminate the program.

9. The **Watch Window** in a debugger is used to monitor the values of specific variables or expressions in real-time during the execution of your program. It allows you to track the values of variables as the program runs.
   The Watch Window is typically empty at the start of a debugging session because automatically adding all variables would consume resources and potentially slow down the debugging process.
   We need to choose which variables or expressions to monitor based on what we are debugging. This allows for more focused and efficient debugging

10. Debugging using Visual Studio Debugger from steps 2-5:

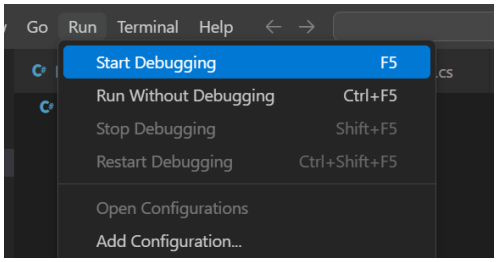
    Activity 2: Break points added (the red dots):



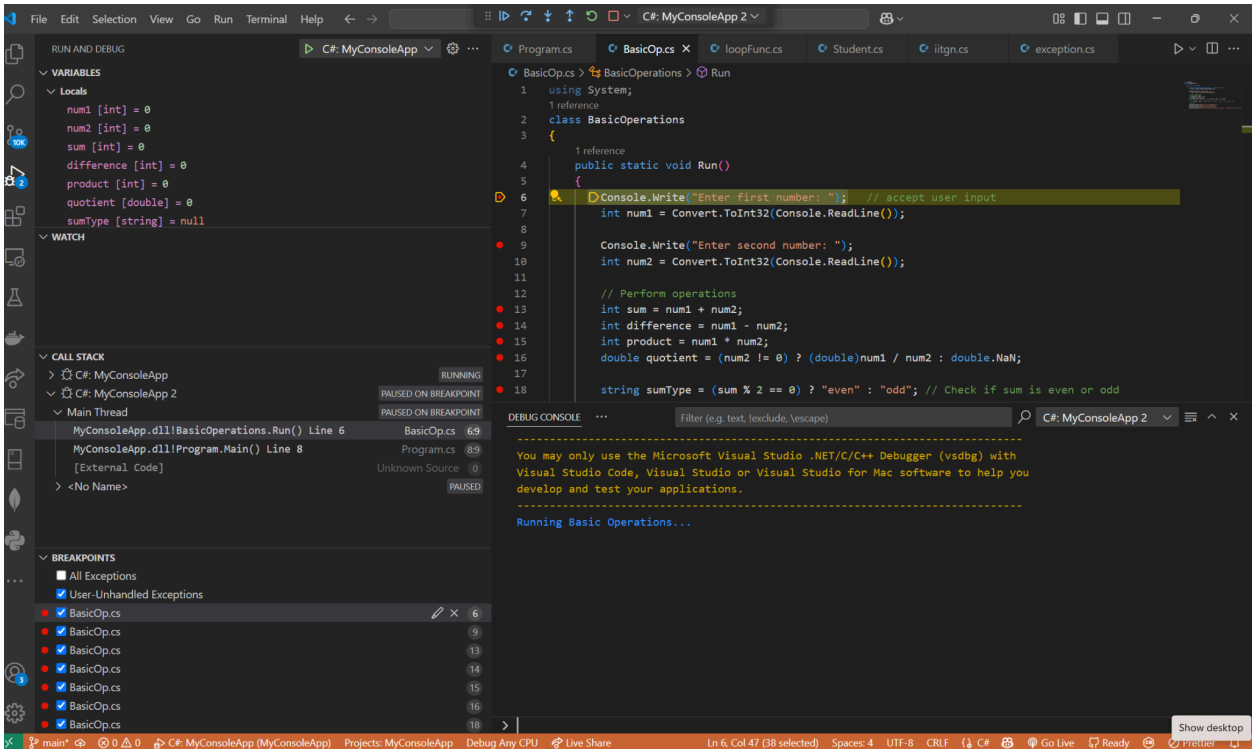
```
BasicOp.cs > BasicOperations > Run
1   using System;
    1 reference
2   class BasicOperations
3   {
        1 reference
4       public static void Run()
5       {
6           Console.Write("Enter first number: ");   // accept user input
7           int num1 = Convert.ToInt32(Console.ReadLine());
8
9           Console.Write("Enter second number: ");
10          int num2 = Convert.ToInt32(Console.ReadLine());
11
12          // Perform operations
13          int sum = num1 + num2;
14          int difference = num1 - num2;
15          int product = num1 * num2;
16          double quotient = (num2 != 0) ? (double)num1 / num2 : double.NaN;
17
18          string sumType = (sum % 2 == 0) ? "even" : "odd"; // Check if sum is even or odd
19
20          // Results
21          Console.WriteLine($"Addition: {sum} ({sumType})");
22          Console.WriteLine($"Subtraction: {difference}");
23          Console.WriteLine($"Multiplication: {product}");
24          Console.WriteLine($"Division: {(num2 != 0 ? quotient.ToString() : "Cannot divide by zero")}");
25      }
26  }
```

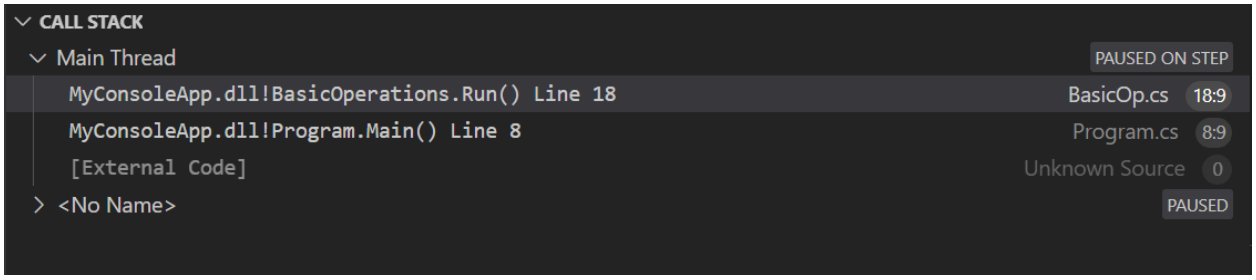
Line	Reason for Breakpoint
6	Inspect `num1` input to check for invalid conversions (e.g., non-numeric input).
13	Verify `num2` input and check for division-by-zero cases.
14	Ensure correct sum calculation (`num1 + num2`).
15	Verify subtraction operation (`num1 - num2`).
16	Check multiplication operation (`num1 * num2`).
18	Debug division logic to prevent `num2 == 0` errors.
21	Confirm `sumType` correctly determines even/odd sum.
22	Inspect addition output with correct even/odd classification.
23	Verify subtraction result output.
24	Debug division output (ensure "Cannot divide by zero" displays properly when needed).

Start debugging:



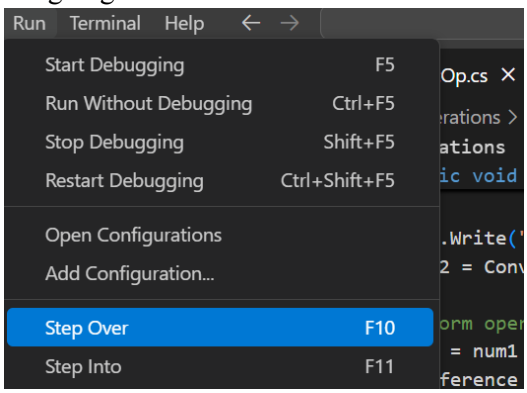

```
C:\Program Files\dotnet\sdk\9.0.202\Microsoft.Common.CurrentVersion.targets(5364,5): error
ERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp\obj\Debug\net9.0\apphost.exe" to "bin\Debug\net9.
ers\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp\bin\Debug\net9.0\MyCon
Users\VENU GOPALROA\Desktop\TERM 2 MATERIALS\3-2\STT\lab9\MyConsoleApp\MyConsoleApp.csproj]

*  The terminal process terminated with exit code: 1.
*  Terminal will be reused by tasks, press any key to close it.
```



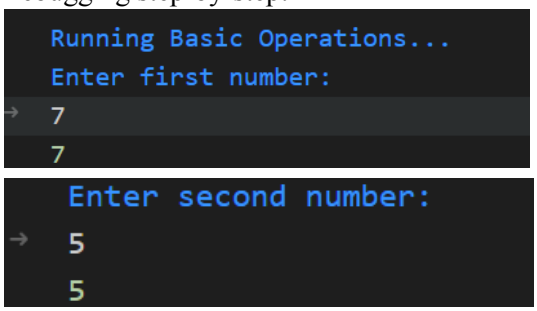


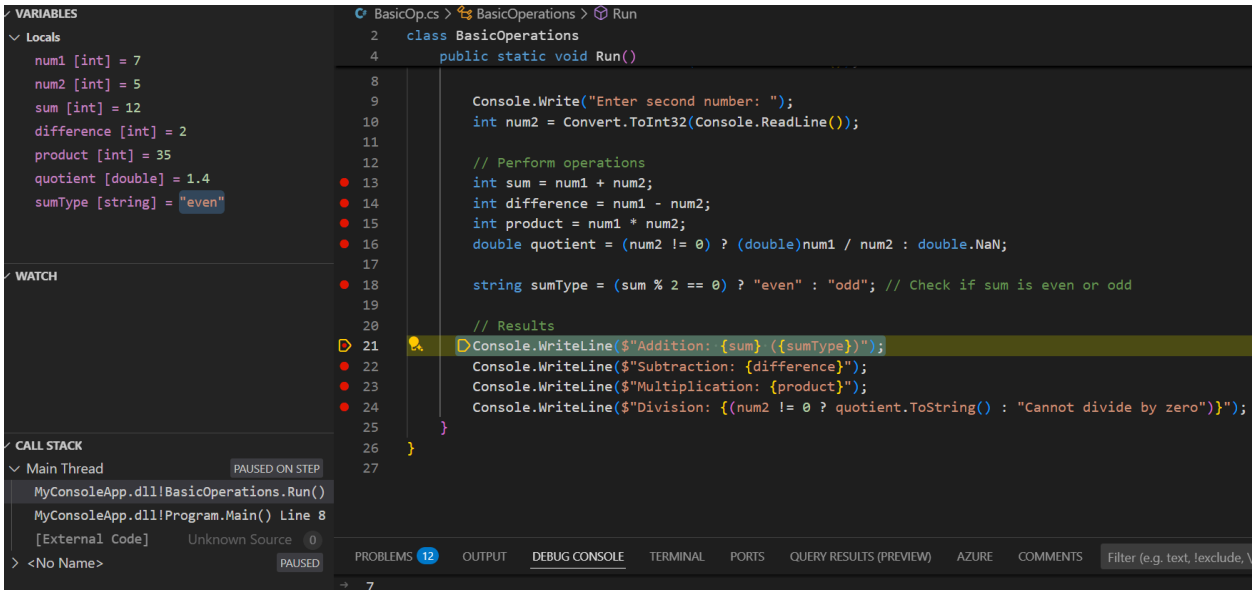
```
static void Main()
{
    Console.WriteLine("Running Basic Operations...");
    BasicOperations.Run();  // Calls the BasicOperations class
    LoopsFunctions.Run();  // Calls the LoopsFunctions class
    Student student = new Student("John Doe", 12345, 85.5);
    student.DisplayDetails();  // Calls the DisplayDetails method of the Student c
    StudentIITGN studentIITGN = new StudentIITGN("Jane Doe", 67890, 92.0, "Hostel
    studentIITGN.DisplayIITGNDetails();  // Calls the DisplayIITGNDetails method o
    Console.WriteLine("\nRunning Exception Handling...");
    ExceptionHandling.Run();
```

For going into next line:

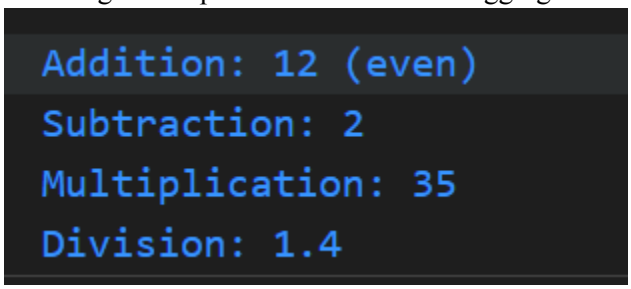


Debugging step-by-step:





Checking the output at each line for debugging.



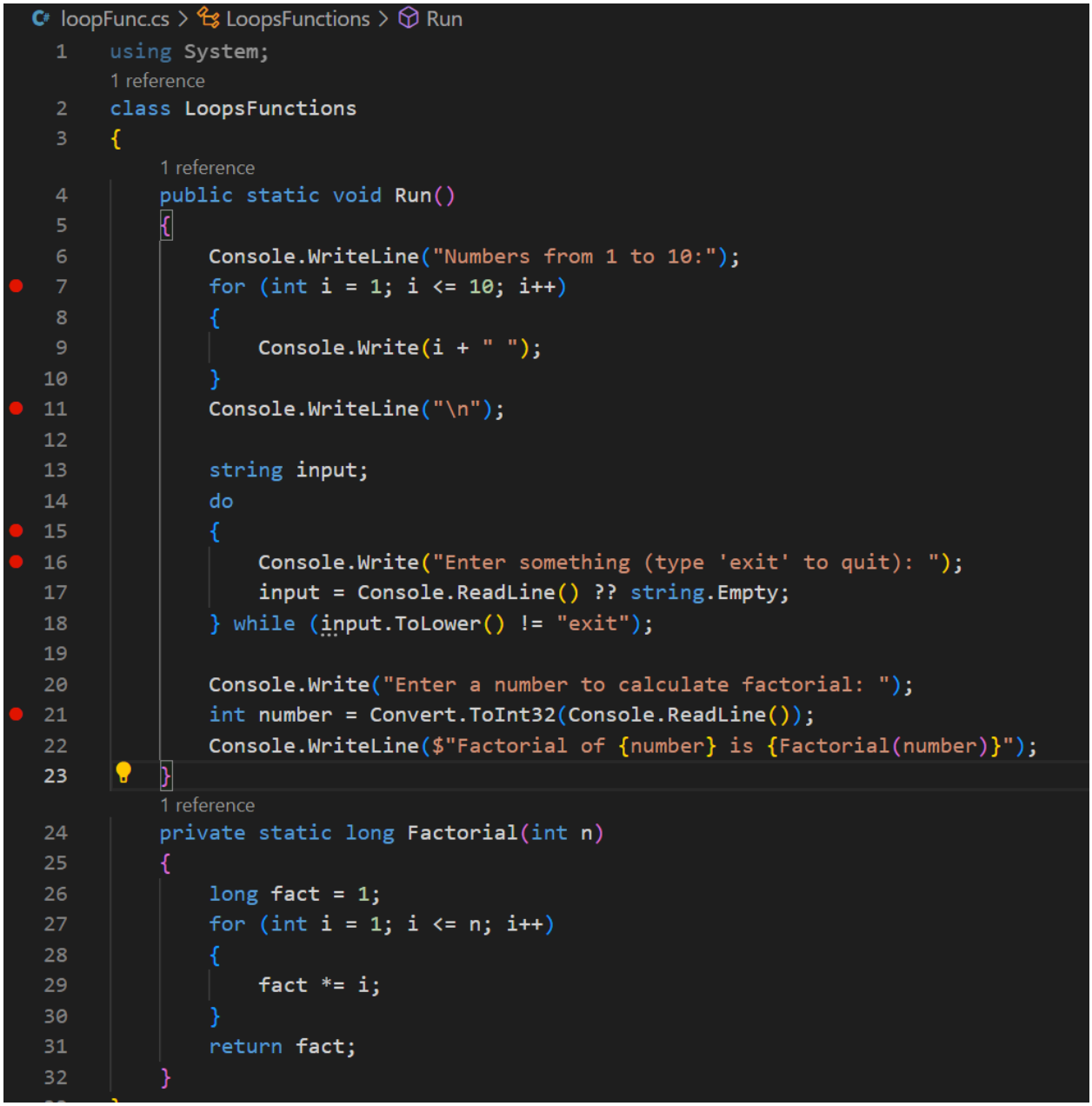
Another example:

```
Enter first number:
To send text to the target process's star
ng. See https://aka.ms/VSCode-CS-LaunchJs
4

Enter second number:
7

Addition: 11 (odd)
Subtraction: -3
Multiplication: 28
Division: 0.5714285714285714
Numbers from 1 to 10:
1 2 3 4 5 6 7 8 9 10
```

11. Activity 3:

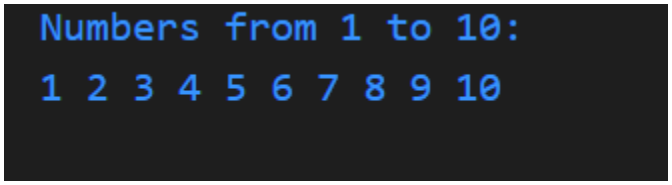
```
9       LoopsFunctions.Run();  // Calls the LoopsFunctions class
10          Student student = new Student("John Doe", 12345, 85.5);
11          student.DisplayDetails();  // Calls the DisplayDetails metho
12          StudentIITGN studentIITGN = new StudentIITGN("Jane Doe", 678
13          studentIITGN.DisplayIITGNDetails();  // Calls the DisplayIIT
14          Console.WriteLine("\nRunning Exception Handling...");
15          ExceptionHandling.Run();
16      }
17  }
```

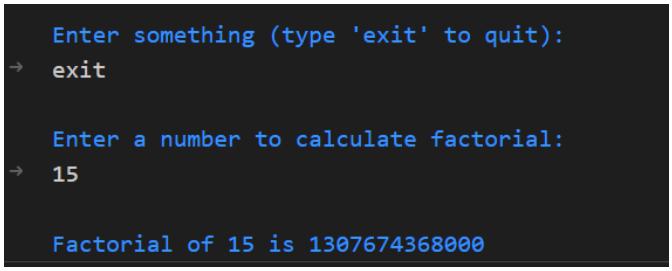
Debug points:



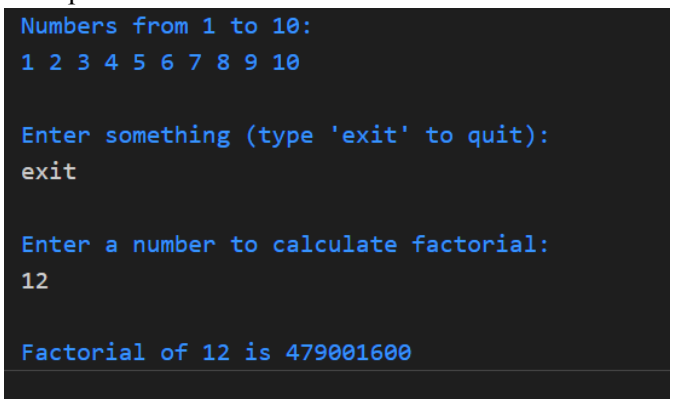
Line	Reason for Breakpoint
7	Check `i` values in for-loop
11	Inspect user input in `do-while` loop
16	Ensure valid number input
17	Debug factorial calculation
22	Track factorial multiplication

OUTPUT:



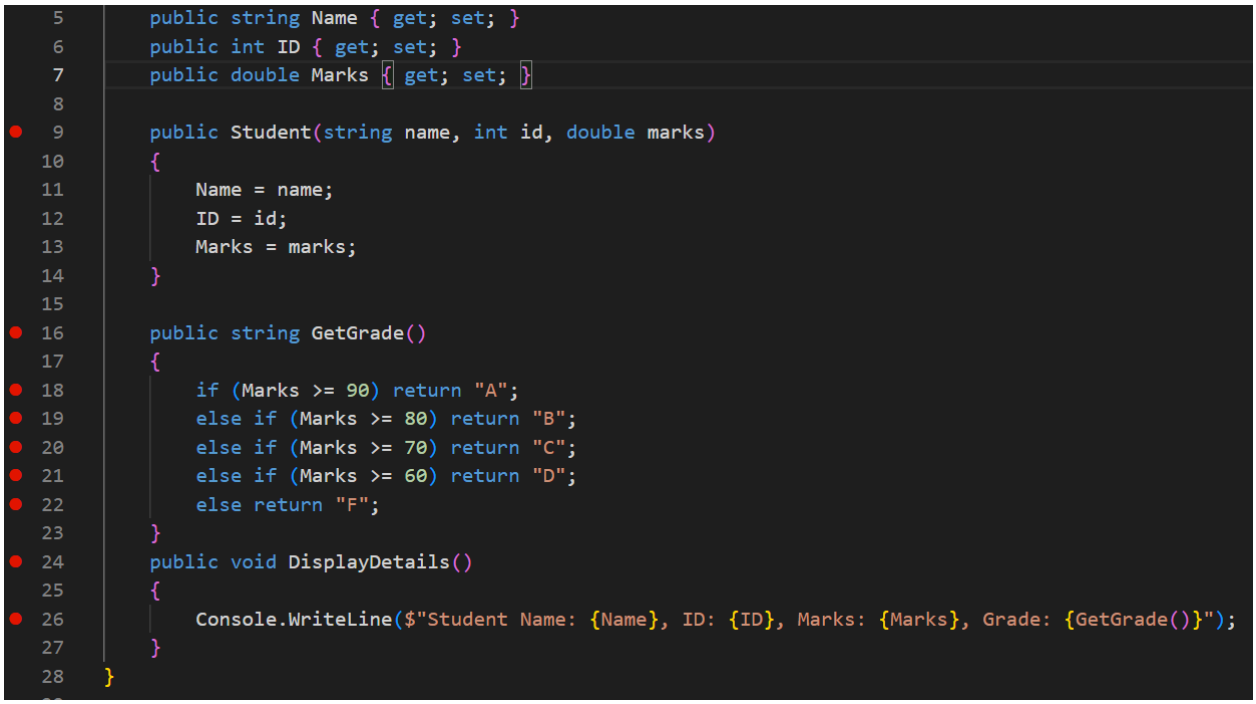


Example 2:



12. **Activity 4:**
Break points:

Line	Reason for Breakpoint
8	Debug constructor execution to ensure correct assignment of `Name`, `ID`, and `Marks`.
14	Verify `GetGrade()` logic to check correct grade assignment.
16	Ensure correct execution of grade logic for marks ≥ 80 but < 90.
18	Check proper execution when marks are in the 70–79 range.
20	Validate condition for marks in the 60–69 range.
22	Ensure `F` grade is assigned correctly for marks < 60.
26	Debug `DisplayDetails()` method to check if all values are printed correctly.
26 *(inside `GetGrade()` call)*	Step into `GetGrade()` method from `DisplayDetails()` to verify grade retrieval.



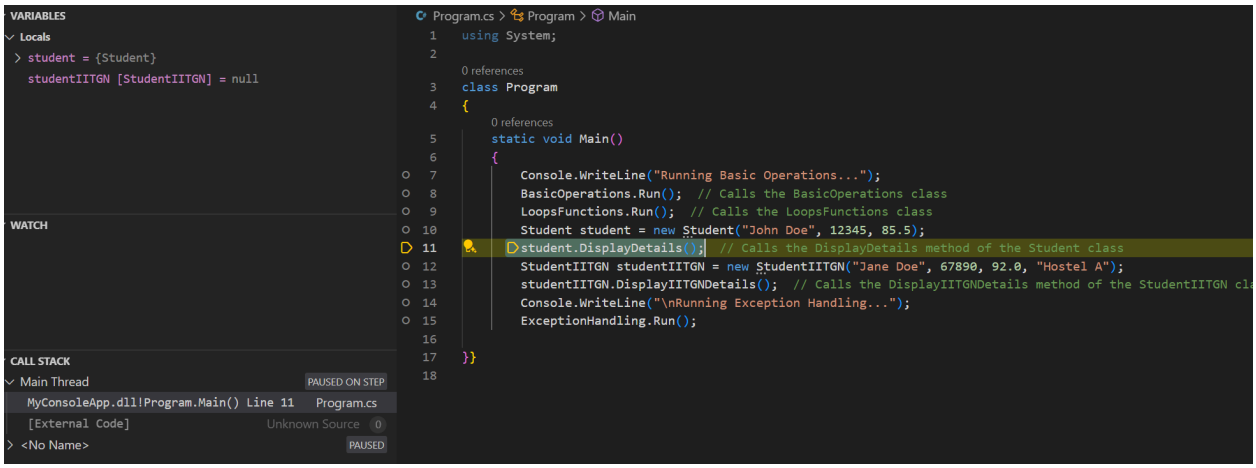
```
5       public string Name { get; set; }
6       public int ID { get; set; }
7       public double Marks { get; set; }
8
9       public Student(string name, int id, double marks)
10      {
11          Name = name;
12          ID = id;
13          Marks = marks;
14      }
15
16      public string GetGrade()
17      {
18          if (Marks >= 90) return "A";
19          else if (Marks >= 80) return "B";
20          else if (Marks >= 70) return "C";
21          else if (Marks >= 60) return "D";
22          else return "F";
23      }
24      public void DisplayDetails()
25      {
26          Console.WriteLine($"Student Name: {Name}, ID: {ID}, Marks: {Marks}, Grade: {GetGrade()}");
27      }
28  }
```

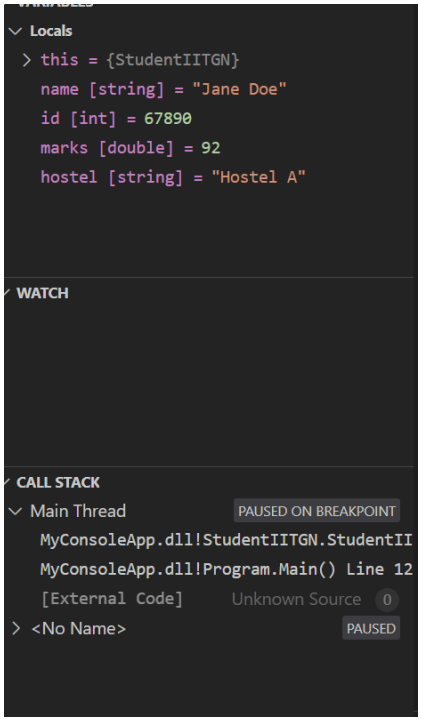
Line Number	Code	Reason for Breakpoint
4	`public StudentIITGN(string name, int id, double marks, string hostel)`	Before the constructor is called to inspect object initialization.
5	`: base(name, id, marks)`	Before the call to the base class constructor to ensure proper inheritance.
8	`Hostel_Name_IITGN = hostel;`	Before setting the `Hostel_Name_IITGN` property to track the value being assigned.
12	`Console.WriteLine($"IITGN Student: {Name}, ID: {ID}, Marks: {Marks}, Grade: {GetGrade()}, Hostel: {Hostel_Name_IITGN}");`	Before displaying the student details to check the output and values.

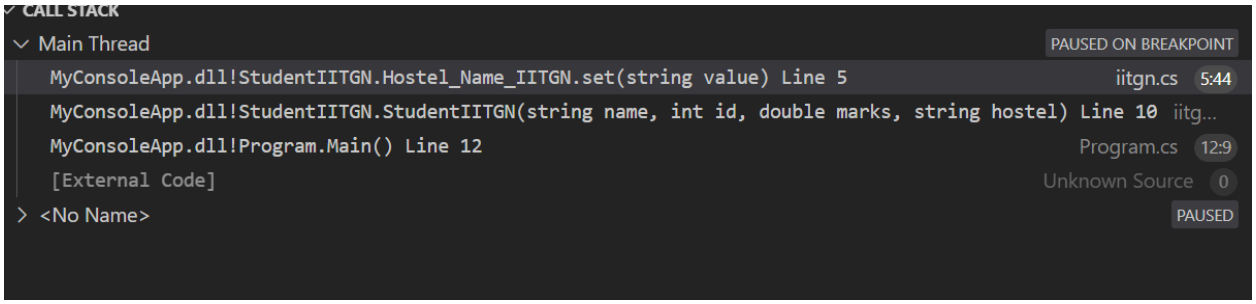
```
1   using System;
2
    3 references
3   class StudentIITGN : Student
4   {
        2 references
5       public string Hostel_Name_IITGN { get; set; }
6
        1 reference
7       public StudentIITGN(string name, int id, double marks, string hostel)
8           : base(name, id, marks)
9       {
10          Hostel_Name_IITGN = hostel;
11      }
12
        1 reference
13      public void DisplayIITGNDetails()
14      {
15          Console.WriteLine($"IITGN Student: {Name}, ID: {ID}, Marks: {Marks},
16      }
```

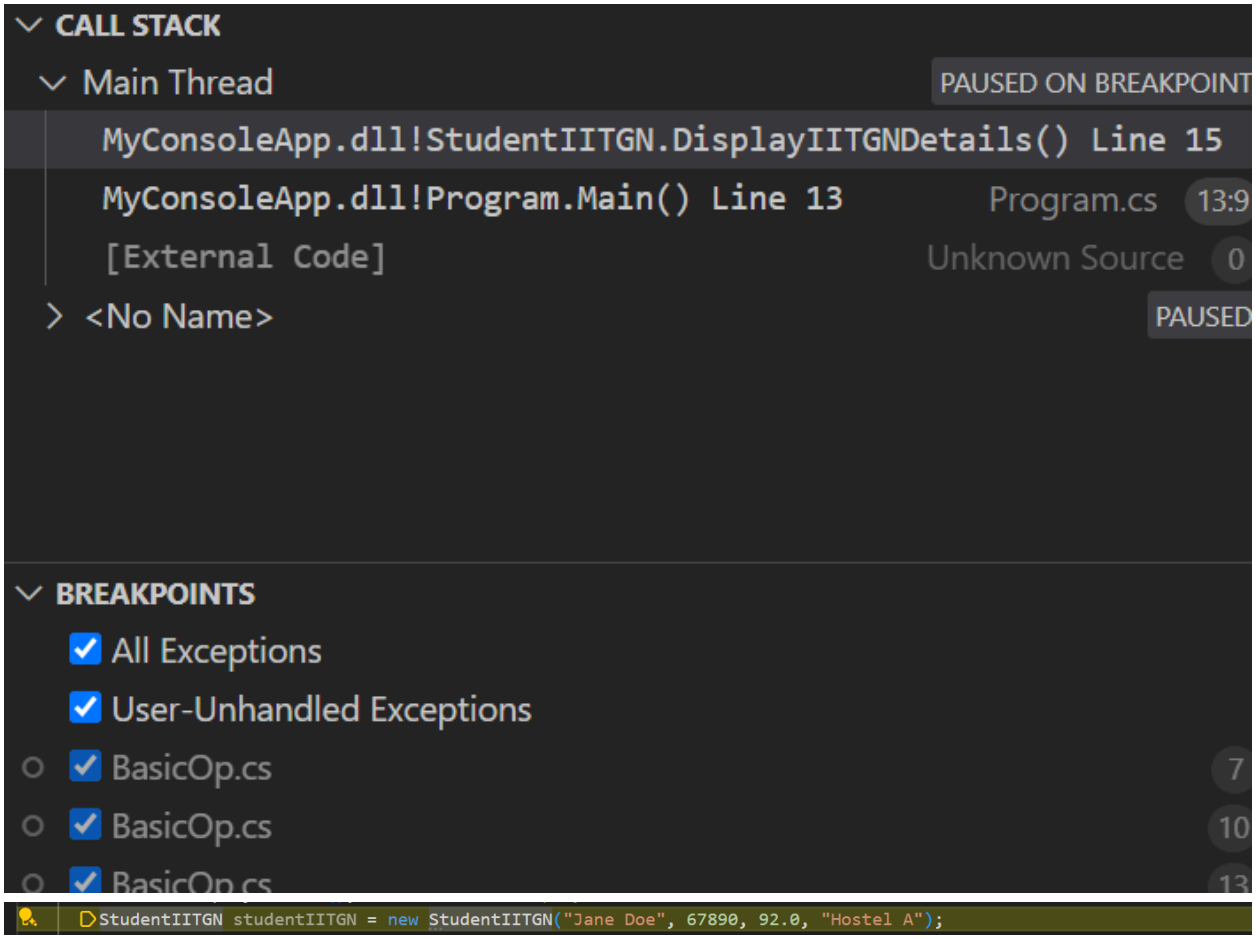
Debugging:

```
    Student student = new Student("John Doe", 12345, 85.5);
    student.DisplayDetails();  // Calls the DisplayDetails method of the Student class
    StudentIITGN studentIITGN = new StudentIITGN("Jane Doe", 67890, 92.0, "Hostel A");
    studentIITGN.DisplayIITGNDetails();  // Calls the DisplayIITGNDetails method of the St
    Console.WriteLine("\nRunning Exception Handling...");
    ExceptionHandling.Run();
}
```






CALL STACK
Main Thread
PAUSED ON BREAKPOINT
MyConsoleApp.dll!StudentIITGN.Hostel_Name_IITGN.set(string value) Line 5
iitgn.cs 5:44
MyConsoleApp.dll!StudentIITGN.StudentIITGN(string name, int id, double marks, string hostel) Line 10
iitg...
MyConsoleApp.dll!Program.Main() Line 12
Program.cs 12:9
[External Code]
Unknown Source 0
<No Name>
PAUSED


CALL STACK
Main Thread
PAUSED ON BREAKPOINT
MyConsoleApp.dll!StudentIITGN.DisplayIITGNDetails() Line 15
MyConsoleApp.dll!Program.Main() Line 13
Program.cs 13:9
[External Code]
Unknown Source 0
<No Name>
PAUSED
BREAKPOINTS
All Exceptions
User-Unhandled Exceptions
BasicOp.cs 7
BasicOp.cs 10
BasicOp.cs 13
StudentIITGN studentIITGN = new StudentIITGN("Jane Doe", 67890, 92.0, "Hostel A");

```
iitgn.cs > StudentIITGN > .ctor
1   using System;
2
    3 references
3   class StudentIITGN : Student
4   {
        2 references
5       public string Hostel_Name_IITGN { get; set; }
6
        1 reference
7       public StudentIITGN(string name, int id, double marks, string hostel)
8           : base(name, id, marks)
9       {
10          Hostel_Name_IITGN = hostel;
11      }
12
        1 reference
13      public void DisplayIITGNDetails()
14      {
15          Console.WriteLine($"IITGN Student: {Name}, ID: {ID}, Marks: {Marks},
```

```
3   class StudentIITGN : Student
4   {
        2 references
5       public string Hostel_Name_IITGN { get; set; }
6
```

```
    StudentIITGN studentIITGN = new StudentIITGN("Jane Doe", 67890, 92.0, "Hostel A");
    studentIITGN.DisplayIITGNDetails();  // Calls the DisplayIITGNDetails method of the StudentIITGN class
```

```
13  public void DisplayIITGNDetails()
14  {
15      Console.WriteLine($"IITGN Student: {Name}, ID: {ID}, Marks: {Marks}, Grade: {GetGrade()}, Hostel: {Hostel_
16  }
```

```
VARIABLES
Locals
  System.Runtime.CompilerServices.DefaultInterpolatedStringHandler.ToStringAndClear returned [string] = "IITGN Student: Jane Doe, ID: 67890, Marks: 92, Grad…
  > this = {StudentIITGN}
```

```
VARIABLES
  Locals
    > student = {Student}
    > studentIITGN = {StudentIITGN}
```

OUTPUT:

```
Student Name: John Doe, ID: 12345, Marks: 85.5, Grade: B
IITGN Student: Jane Doe, ID: 67890, Marks: 92, Grade: A, Hostel: Hostel A
```

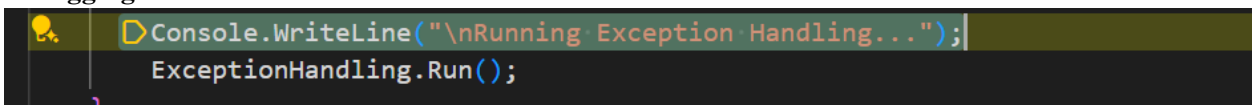
13. Activity 5:

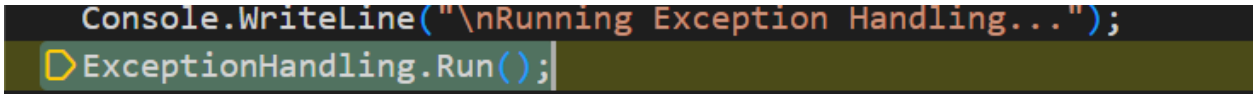
Break points:

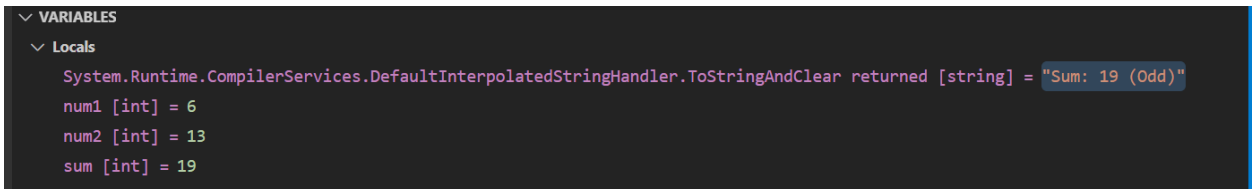
Line Number	Code	Reason for Breakpoint
6	`Console.Write("Enter first number: ");`	Before reading the first number to inspect input behavior.
7	`int num1 = Convert.ToInt32(Console.ReadLine());`	Before converting the input to an integer to catch any potential `FormatException`.
10	`Console.Write("Enter second number: ");`	Before reading the second number to inspect input behavior.
11	`int num2 = Convert.ToInt32(Console.ReadLine());`	Before converting the input to an integer to catch any potential `FormatException`.
14	`int sum = num1 + num2;`	Before performing arithmetic to check if the sum calculation is correct.
15	`Console.WriteLine($"Sum: {sum} ({(sum % 2 == 0 ? "Even" : "Odd")})");`	Before printing the sum and checking whether it is even or odd.
16	`Console.WriteLine($"Multiplication: {num1 * num2}");`	Before displaying the multiplication result to check the arithmetic.
17	`Console.WriteLine(num2 != 0 ? $"Division: {num1 / num2}" : "Cannot divide by zero");`	Before performing division to inspect if division by zero is handled.
19	`catch (FormatException)`	Before catching a `FormatException` to examine the specific exception thrown for invalid input.
21	`catch (Exception ex)`	Before catching other exceptions to inspect what error occurred.

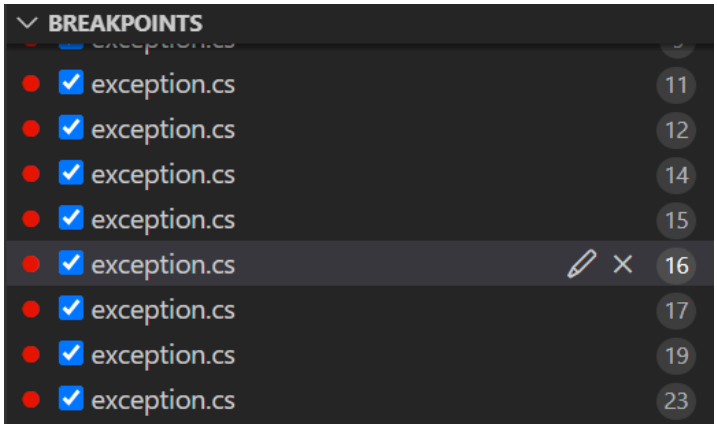
```
4   public static void Run()
5     {
6         try
7         {
8             Console.Write("Enter first number: ");
9             int num1 = Convert.ToInt32(Console.ReadLine());
10
11            Console.Write("Enter second number: ");
12            int num2 = Convert.ToInt32(Console.ReadLine());
13
14            int sum = num1 + num2;
15            Console.WriteLine($"Sum: {sum} ({(sum % 2 == 0 ? "Even" : "Odd")})");
16            Console.WriteLine($"Multiplication: {num1 * num2}");
17            Console.WriteLine(num2 != 0 ? $"Division: {num1 / num2}" : "Cannot divide by zero");
18        }
19        catch (FormatException)
20        {
21            Console.WriteLine("Invalid input! Please enter a number.");
22        }
23        catch (Exception ex)
24        {
25            Console.WriteLine($"Error: {ex.Message}");
26        }
```

Debugging:










```
14              int sum = num1 + num2;
15              Console.WriteLine($"Sum: {sum} ({(sum % 2 == 0 ? "Even" : "Odd")})");
16              Console.WriteLine($"Multiplication: {num1 * num2}");
17              Console.WriteLine(num2 != 0 ? $"Division: {num1 / num2}" : "Cannot divide by zero");
18          }
```

```
17              Console.WriteLine(num2 != 0 ? $"Division: {num1 / num2}" : "Cannot divide by zero");
18          }
19          catch (FormatException)
20          {
21              Console.WriteLine("Invalid input! Please enter a number.");
22          }
```

OUTPUT:

```
Running Exception Handling...
Sum: 19 (Odd)

Running Exception Handling...
Multiplication: 78
```

Exception handling if we give a string instead of string for factorial calculation:
Here I entered a string "exit" instead of a number to calculate factorial.

```
Enter a number to calculate factorial:
exit


Running Exception Handling...

Running Exception Handling...
Unhandled exception. System.FormatException: The input string 'exit' was not in a correct format.
```

When the denominator is zero:

```
Running Exception Handling...
Enter first number:
10

Enter second number:
0

Sum: 10 (Even)
Multiplication: 0
Cannot divide by zero
The program '[30036] MyConsoleApp.exe' has exited with code 0 (0x0).
Please start a debug session to evaluate expressions
```

```
Division: Cannot divide by zero
```

When the input is not an int:

```
Locals
> $exception [FormatException] = {System.FormatException: The input string 'hii' was not in a correct format.
  num1 [int] = 6
  num2 [int] = 0
  sum [int] = 0
  difference [int] = 0
  product [int] = 0
  quotient [double] = 0
  sumType [string] = null
```

Similarly, we get the "Invalud input! Please enter a number." in the debug console, when we enter a string instead of number.

The debugging is done for all activities:

```
The program '[20244] MyConsoleApp.exe' has exited with code 0 (0x0).
```

Reflections and Lessons Learned:

- Error Handling: Implementing exception handling is crucial to prevent crashes and ensure a smooth user experience.
- Debugger Utility: The Visual Studio Debugger significantly helps in tracing program execution and identifying issues.
- OOP Principles: Understanding inheritance and method overriding was essential for extending classes.
- Loops and Conditionals: Essential control structures that form the foundation of C# programming.

Summary:

This lab provided me hands-on experience in developing C# console applications. By implementing various programming constructs and debugging techniques, I have gained a good understanding of .NET

development. The structured approach to handling input, processing data, and debugging ensures efficient and error-free coding practices in C#.

Resources:

- https://drive.google.com/file/d/1bJG8oi3f2s-XBT3OZEEZ0idv-B4lzSAA/view
- https://drive.google.com/file/d/12-KtkebmYLFGosvB3tyHNEVQYe--tkbV/view
- https://learn.microsoft.com/en-us/dotnet/csharp/
- https://github.com/dotnet/dotnet-console-games
- http://visualstudio.microsoft.com/
- https://dotnet.microsoft.com/en-us/download
- https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/exceptions/
- https://learn.microsoft.com/en-us/visualstudio/debugger/?view=vs-2022
- https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/inheritance