

CSC 236 A02: OOP Principles

Enter your name:

Some rules and procedures THAT YOU NEED TO READ (at least once):

- This assignment is to be done individually, although you may discuss the work with the professors, the TAs, and even your classmates.
Any copies of your work or allowing anyone else to copy your work is academic dishonesty and will be treated as such.
- Make a copy of this worksheet in your private Google account.
- Change the document title username to be your Berea username.

Objective:

The purpose of this assignment is to deepen our understanding of OOP Principles by looking at OOP through a different lens. You will first examine some articles and blogs that explain OOP in different ways. Then you will synthesize what you have read of OOP principles into a reasoned argument about whether OOP or procedural programming is better in general or whether they each have their purposes.

Part 1:

1. Read the following articles/blogs:
 - [The Story of Object-Oriented Programming](#)
 - [Advantages of OOP | Explore the Top 9 Advantages of OOP](#)
 - [What is Procedural Programming? - Key Features](#)
2. **Using only a search tool like Google**, make an alphabetized glossary of some of the **key technical terms** in the three linked articles. (We have given you a good start below- you must do at least the ones listed, but you may add more.) You are likely to have to search for each of these terms because the articles do not define many of them but instead give insights and analogies.

Write each term, definition, and source URL from each of the articles in alphabetical order in the following table using the "technical term: definition - source" structure. We have given OOP as an example.

Note: it is perfectly okay to quote directly for this section. Be sure to give the reference where you found the term. Do not use AI, use only a search tool.

Glossary

Example:

- **Object-Oriented Programming (OOP)** - “Object-oriented programming (OOP) is a programming language model organized around objects rather than actions and data rather than (programming) logic.” Quoted from: [What is Object-Oriented Programming \(OOP\)? | Definition from TechTarget.](#)

*Note that there are **a lot of these**— if you add any, the bullets will expand when you push enter.*

- Attribute:
- Class:
- Functional Programming:
- Function call:
- Global variable:
- Hierarchy:
- Immutable data:
- Imperative programming model:
- Mutable data:
- Object:
- Object behavior:
- Override:
- Paradigm:
- Parallel programming:
- Parameter passing:
- Plugin:
- Predefined functions:
- Procedural programming:
- Reference:
- Source code:
- Subclass:

3. Write a more extensive description for each of the four key principles of OOP, namely Abstraction, Encapsulation, Inheritance, and Polymorphism. You should aim for paragraphs of 5-7 sentences per concept. This time DO NOT quote, but instead paraphrase so the paragraph you write is entirely your own words. **Do NOT use AI for this—you need to build your own understanding.**

a. Abstraction

b. Encapsulation

c. Inheritance

d. Polymorphism

Part 2:

First, a philosophical interlude on learning programming paradigms

College is the typical place where you can learn how to learn. This ability will help you to grow into a respected expert.

While the Internet now contains a large diversity of viewpoints, some are reasonable and valid, but some are utter garbage that are written entirely to earn money via number of clicks (such inflammatory yet highly inaccurate articles are often called “clickbait.”). Unfortunately, disinformation on the Internet has become a major problem. We hope you will learn to distinguish between these reputable articles and clickbait, even in the situations when real experts disagree (and are not writing garbage designed as clickbait.)

OOP is the first topic in this course where you can find real experts disagreeing about the importance of OOP versus procedural programming, which is what you primarily focused on in CSC226. We expect you to know what both procedural programming and OOP are as well as to be able to program in OOP regardless of what you ultimately decide that you think about it as a paradigm. Why? Because you should never adopt nor reject any programming paradigm if you do not understand it sufficiently well to both explain and defend your position.

Secondly, what you should do.

We expect you to fairly present OOP vs procedural programming in the remainder of this assignment as if you were discussing this topic with a job recruiter.

4. In at least two paragraphs, write a short essay **fairly** presenting both of the paradigms of OOP vs procedural programming. You **must both** read and use all of the articles we linked above, but you may optionally add other reputable references as additional references as long as you truly see them as actually reputable. **Again, do NOT use AI for this—you need to build your own understanding.**

If you do add other articles, include their references using APA format¹ and also add a single sentence WHY the reference is reputable rather than clickbait.

5. What was the most significant thing you learned (or re-remembered) that was covered in the linked articles above while writing the essay?

¹ see [Webpage on a website references](#)

6. What is your overall reaction to this assignment? Do you have any recommendations for improvement?

Integrity statement

Please briefly describe ALL help you received and all help you gave to others in completing this assignment.

To Submit:

- Download this document as a pdf and submit it to Moodle by the due date.