CSE 210: Programming with Classes

Intructor: Nathan Parrish

October 10, 2024

**What is abstraction and why is it important?**

Abstraction refers to the principle of breaking your program or function into fragments or pieces, which can then be used to compile into one program. There are several benefits of using abstraction when programming including allowing the developer the opportunity to visualize their program as they plan attributes and behaviors of each class. Another benefit is that it reduces bugs, since every aspect of the program is accounted for, making testing and debugging a lot easier. It also increases the readability of your code and adds a level of security since it will only show certain data and allow access and modification through classes and methods. An example of this would be in my Journal Entry program. I used private attributes, which helped to prevent my class variables from being accessed outside of that class, I had to use methods as a way of accessing these variables by using the concept of getters and setters. Ex.:

User.cs

**public class User**

**{**

**private string \_userName;**

**public User()**

**{**

**\_userName = GetUserName();**

**// Set username as an object**

**}**

**public string GetUserName()**

**{**

**Console.Write("Please enter your name: ");**

**return Console.ReadLine();**

**// Returns the inputted name**

**}**

**public void SetUserName(string userName)**

**{**

**\_userName = userName;  // Set the private variable to the passed-in value**

**}**

**public void DisplayUserName()**

**{**

**Console.WriteLine($"Hello, {\_userName}, welcome to the e-journal!");**

**}**

**}**

With the above code, I cannot access the attributes from outside the class since the variable “**\_userName”** is set to private. To access it from outside the class, I had to create an instance of the class and assign it to a new object to read the stored variables this increases the security since the variables are private. Ex.:

Program.cs

**// Create an instance of the User User user = new User();**

**user.DisplayUserName(); // Display the user's name**