**Abstractions**

* Explain the meaning of abstraction?

Abstractions refers to the act of representing essential features without including the background details or explanation.

* Highlight a benefit of abstraction?

Reusability of the data.

As a developer develops program for companies most at time, the companies’ owners are the only people who have access to the code and know the inners functions of the code, with data abstraction consumers become unaware of the technical details and companies can us that data without disrupting how consumers use their product. With data abstraction owners of the companies with the aid of a developer can create an object of the same kind again.

* Provide an application of abstraction.

An application of abstraction could be in the banking system. Security technologies use data abstraction to safeguard customers sensitive banking information. At an ATM, customers know to slide their debit cards through the slots and enter their credentials to withdraw money from their accounts. They also know how to tailor their deposits from checking or savings accounts and view their balance after the transaction.

Due to data abstraction, customers cannot see how the ATM dispenses money from the inside to the slot on the outside, allowing the customers to grab the bills. They also cannot tell how the ATM provides a printed receipt of the withdrawal. Therefore, the details of the action remain private, preventing outsiders from access their funds.

* Use a code example Abstraction from the program wrote
* using System.IO;
* using System.Collections.Generic;
* public class Journal
* {
* public List<string> \_prompts = new List<string>()
* {"What was the best part of your day? ",
* "What is your favorite food? ",
* "What is something you learned these past few weeks? ",
* "What is something that motivates you to do your best every day? "};
* public List<Entry> \_entries = new List<Entry>();
* public Journal()
* {
* }
* public void Write()
* {
* Random rnd = new Random();
* int num = rnd.Next(\_prompts.Count);
* string prompt = \_prompts[num];
* Console.WriteLine(prompt);
* string response = Console.ReadLine();
* DateTime theCurrentTime = DateTime.Now;
* string dateText = theCurrentTime.ToShortDateString();
* Entry entry = new Entry();
* entry.\_dateTime = dateText;
* entry.\_prompt = prompt;
* entry.\_message = response;
* \_entries.Add(entry);
* }
* public void DisplayIt()
* {
* foreach(Entry entry in  \_entries)
* {
* entry.Display();
* }
* }
* public void Load()
* {
* Console.WriteLine("What is the filename? ");
* string file = Console.ReadLine();
* using (StreamReader sr = new StreamReader(file))
* {
* // Read the stream to a string, and write the string to the console
* string line = sr.ReadToEnd();
* Console.WriteLine(line);
* }
* }
* public void Save()
* {
* Console.WriteLine("to which file? ");
* string file = Console.ReadLine();
* using (StreamWriter outputFile = new StreamWriter(file))
* {
* foreach (Entry entry in \_entries)
* {
* outputFile.WriteLine($"Date: {entry.\_dateTime} - {entry.\_prompt}");
* outputFile.WriteLine($"{entry.\_message}");
* }
* }
* }
* }