Austin Henthorne

MEC 510: HW1

Object Oriented Programming

Traditional procedural programming suffers from disorganization when a program becomes too big. Object oriented programing (OOP) was created to decrease the complexity of a program. Two main problems that programmers ran into with procedural programming were that functions have unrestricted access to global data. Also, unrelated functions and data were a poor model of the real world. OOP was also evolved to put together *data* and *functions*, which creates an important component of this language which is an *object.* *Objects* are made to help the programmer reflect the real world by dividing the problem into objects, not functions. For example, people or vehicles could be an *object*, which would have their own *behaviors* and *attributes*. Objects also offer the programmer to create new data types. Objects are said to be members of *classes* in OOP. A *class* consists of usually more than one object and serves as a template for those objects. OOP also uses the concept of *inheritance*, which allows the creation of derived classes from the parent class. These derived classes still contain the data and functions of its parent class, but have the flexibility to add new ones of its own. Another important element of OOP is *polymorphism*; the use of operators or functions in different ways, depending on what they are operating on. Adding two objects together is an example of this. In conclusion, object oriented program was created to keep programs more organized and is a proven design.