EXERCISE 1: EVERYDAY OBJECT

Assigned: **Sept. 22**

Due: **Sept. 27**

OBJECTIVES

The objectives of this exercise:

- Computational:
  - Learning about the description and design process for modular programming by describing an everyday object in detail including why the object is needed and how the object functions
  - Learning about abstraction and function characterization by identifying properties of an everyday object
- Creative:
  - Surrounding (looking at an everyday object in new ways, using all of your senses to understand how it's made and how it function)
  - Capturing (using written language to describe all the different details and characteristics of this everyday object so you can work with it in new ways)
  - Challenging (describing the operations of an everyday object with words and also as a computer program)
  - Broadening (imagining that this everyday object doesn't exist and acting like its inventor who is trying to fulfill a need by creating something new and useful)

PROBLEM DESCRIPTION

For this assignment, you will be using language to try to clearly and thoroughly describe the functions of an ordinary object that you might use every day. You will be acting like the inventor of that object, imagining that it does *not* yet exist and trying to describe what need would be fulfilled by your (new) object and how (specifically) it will function.

Each group will submit a written report.

Your group will **choose** a common, functional, everyday object from the list given below (Appendix A). Your challenge is to imagine that this object does not exist and to describe in written language (1) the mechanical function of your object, (2) what need is fulfilled by this object, and (3) the physical attributes and characteristics of your object.

You must describe the object's function, the need it will fulfill and its attributes in clear, non-technical language that any user could understand. Your description must be specific enough so that someone who had never seen the object could recognize it and understand how it works and understand what benefits it provides.

For example, if your object is a "colander" you might *begin* to describe it as *"a circular object, approximately 12" in diameter and 9" in height, made of metal or heat-resistant plastic, which is used in cooking to drain pasta after cooking or to hold food for washing or steaming. Its holes are large enough for water and other liquids to drain but small enough so that food will not leak through. A base or foot enables it to sit on a counter or in a sink and handles allow easy carrying and a means to suspend it over a cooking pot for steaming ..."*

This description process is very important for developing algorithms in computer science. An algorithm consists of the series of steps necessary to solve a given problem. By using algorithms, we can solve problems without having to constantly "reinvent the wheel" and spend the time, money, etc. to figure out each step ourselves. However, if any of these steps are unclear, we can have difficulty following the algorithm which can lead to serious repercussions. For example, if the formulation algorithm used to mix the concrete for a road or bridge is unclear, workers may make a mistake during pouring leading to reduced service life. Or, if the business plan algorithm for a new company is confusing, venture capitalists may be reluctant to invest leading to failure of the business. To avoid these repercussions, the developer should make every effort to make the algorithm's description as clear as possible for all steps. In other words, characterization of processes is key; it allows us to abstract a process and then convert it into a formal problem or solution.

1. PART ONE [20 POINTS]

1.1. WRITTEN DESCRIPTION

Over the weekend, generate your written description of your object. Your description must include the following:

1. The mechanical **function(s)**/use(s) of the object (E.g., "This object, which I call a "hammer" is used to drive nails into wood or other materials . . .)
2. What need(s) the object fulfills (E.g., Instead of using a brick to drive nails, the hammer . . .)
3. The **physical attributes** and **physical characteristics** of the object. These include:
   - components or parts (E.g., "The hammer has a handle and a head. The head may have a curved claw like end so that nails can be removed . . . ")

- shape or materials (E.g., “The head is metal. The handle may be wood or metal and may have rubber padding . . .”)
- general dimensions (E.g., “The hammer may range in length from . . .”)
- connections between parts (E.g., Positions of parts such as inside, outside, top, bottom or relationships between parts, such as fixed, fitted, detaches, swivels, etc.)

Your description *should start with the name of the object* and must have a **minimum** of 150 words. You should have a minimum of 6 attributes for your object. The more functions, needs, and attributes you include the more points the group will receive. Keep in mind that attributes may involve all of your senses. Remember, to receive any points you must have contributed to the description of the object by writing or editing the description on the wiki page.

This written description is very important for writing functions in computer science. Functions are blocks of code written to perform a discrete task. With functions we do not need to repeat the same blocks of code multiple times in the same program. This makes the source code more organized and also makes future changes to the code easier (and less error prone) since we only have to change a function once rather than updating each block separately. When you first start programming, you can get away with writing functions in an ad hoc manner while coding. However, for larger programs, and when working with a group, a written description is critical to making sure all the functions are written correctly. For example, imagine writing all the functions necessary for the F-22 Raptor jet fighter which consists of about 1.7 million lines without starting from a detailed description.

2. Part Two [20 points]

2.1. Analysis and Reflection

Post your responses to these questions using the same wiki page you created during week one.

You are expected to discuss these analysis and reflection questions among your group. In order to receive individual credit for Part 2, each group member must contribute on the answers to these questions. Group members who do not contribute to the wiki page will not receive any points for Part 2.

2.2. Analysis [10 points]

Analysis 1 [5 points]. Consider your object as a computer program. Draw a diagram that shows all its functions as boxes (name them), and for

This diagraming process is important for problem analysis in computer science particularly and in all problem solving in general. Just as we have organized similar blocks of code using functions, we can organize functions with similar inputs and outputs together. This process provides a “big picture” view of the program which is vitally important for initial development of the code and future changes. For example, software for large insurance companies may contain many similar functions used for different insurance plans all of which need to be updated after a law is changed.

each function, its inputs and outputs. Are there shared inputs and outputs among the functions? Discuss.

Analysis 2 [5 points]. Consider the list of physical attributes and characteristics. Organize these such that each is declared as a variable with its proper type. Can some of these attributes/characteristics be arranged into a hierarchy of related attributes/characteristics? Discuss.

2.3. REFLECTION [10 POINTS]

Reflection 1 [5 points]. Considering your response to Analysis 1, are there functions that can be combined so that the object can be represented with a more concise program? Are there new functions that should be introduced to better describe your object such that the functions are more modular? Discuss.

Reflection 2 [5 points]. Have you heard of abstraction? How does abstraction in computer science relate to the process of identifying the functions and characteristics as you have done in this exercise? Discuss.

This abstraction process is used in many programming languages to allow similar functions to be written more concisely, and to be more easily understood in a conceptual way. The basic idea is to write the source code completely for only one function in such a group. The rest of the functions use this function as a baseline adding only the source code necessary for their specific tasks. In this way, source code common to multiple functions needs to be written only once. Again, the main advantage is in terms of organization—including defining the relationships between functions—and making updates to the functions. For example, in a simulation game, you could have hundreds of functions for customizing character appearance. By using abstraction, you can avoid having to update all hundreds of functions when you change the common source code on character appearance.

DEADLINES AND HAND-IN

Week 1 Deadline – [Sept. 27, 9:30 a.m.]: You should have completed the description of the object by the Week 1 deadline above. This description should be posted to Canvas.

Week 2 Deadline – [Sept. 29, 9:30 a.m.]: Your analysis and reflection responses are due by the Week 2 deadline above. These responses should be posted to Canvas.

APPENDIX A. LIST OF OBJECTS

List of objects:

zipper
mechanical pencil
binder clip
ziploc bag
scissors
tape measure
stapler
nail clippers
umbrella
flashlight
can opener
clothespin
sticky notes (Post-Its)
toilet paper holder
revolving door
computer mouse
pliers
ball point pen
mousetrap
screwdriver
pocket calculator
sundial
belt
solid air freshener

APPENDIX B.

The US Patent Database, uspto.gov, has examples of how common objects were described for patent purposes.

You can view the original patent application for masking tape and its extension, Scotch tape (Patent 1,760,820; May 27, 1930) at
http://patimg1.uspto.gov/.piw?Docid=1760820&idkey=NONE

Your described object should be able to meet the requirements of a “utility patent.” That is it is new, useful, functions as described, is non-obvious, and is not simply a combination of other existing inventions or a remaking of an existing object.