COURSE NAME

CSCE XXX Exercise X: Everyday Object

Assigned: **XXX**

Due: **XXX**

# Objectives

The objectives of this exercise:

* Computational:
  + Practice *Decomposition* by breaking down a comprehensive description of an object into detailed descriptions of (1) its function(s), (2) the need(s) it fulfills and (3) its physical attributes.
  + Practice *Abstraction* by describing a generic example of an everyday object by focusing on its essential or typical functions and physical attributes without regard to trivial variations (such as color or other variations).
  + Practice *Algorithmic Thinking* by logically, methodically and completely describing an everyday object in sufficient detail and in clear, non-technical language.
  + Practice *Evaluation* by judging the clarity and completeness of the description of an everyday object such that any reader could recognize the object and understand how it works.
  + Describe and design a process for modular programming by describing an everyday object in detail including why the object is needed and how the object functions.
  + Carry out function characterization by identifying properties of an everyday object.
  + Identify and specify input, output, and function of a module or object clearly.
  + Identify details of the inner workings of an object to hide (to not describe) without sacrificing the descriptive clarity of the object’s functionality.
* Creative:
  + Use *Surrounding* by looking at a familiar object in new ways, using all of your senses to understand how it is made and how it functions.
  + Use *Capturing* by using written language to describe all the different details and characteristics of this everyday object so you can work with it in new ways.
  + Use *Challenging* by describing the operations of an everyday object with words and also as a computer program.
  + Use *Broadening* by 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.
* Collaborative/Process:
  + Collaborate: Contributing substantively to the group process, using your skills, knowledge and experience; Being open to all points of view and resolving group conflicts in a constructive way; working together as a team to achieve a common goal; being able to both compete against and cooperate with other teams.
  + Communicate: Giving and receiving thoughtful and constructive feedback in order to develop your group project; valuing others’ contributions and treating them with respect.
  + Coordinate: Meeting group deadlines, including completing your individual work in a timely manner and following the required format.
  + Persist: Being thoughtful and thorough and pushing beyond the easy or conventional response; being willing to test, debug and revise solutions.
  + Play: Imaging, experimenting, taking risks. Being engaged, passionate and open to inspiration from diverse sources; generating multiple approaches before converging on a solution.

# Problem Description

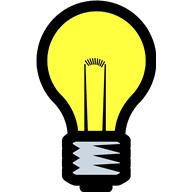
For the next two weeks, you will be using language to try to clearly and thoroughly describe the functions of an ordinary object that you might use every day. You will be acting like the inventor of that object, imagining that it does *not* yet exist and describing what need would be fulfilled by your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

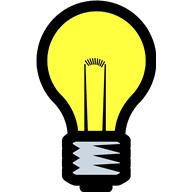
Each group will use a group wiki (visible only to your group) to develop the description of your everyday object in Week 1. In Week 2 each group will use its group discussion board to do a written analysis of and reflection on the description process.

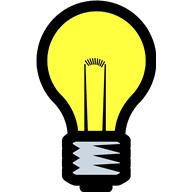
Your group will **choose** a common, functional, everyday object from the list in Appendix A. Your challenge is to imagine that this object does not exist and to describe in written language (1) the mechanical function of your object, (2) what need is fulfilled by this object, and (3) the physical attributes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

You must describe the object’s function, the need it will fulfill and its physical 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” (a strainer for food) 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 drain liquid from other foods 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 ...”*

Communication is an important collaborative and process skill because communicating clearly is essential for anyone in any field. Communication goes beyond sharing ideas and giving and getting feedback. It also includes all the forms of documentation and persuasion that accompany any idea. How will you describe something that hasn’t been seen before? Something that may not even exist yet? To get funding or find partners or investors or market your product you’ll have to describe something your audience hasn’t seen before and you will need to clearly communicate its essential characteristics and its functions or benefits. In a way, effective communication also requires computational thinking skills such as decomposition and pattern recognition. For example, when describing a particular object or product, how would one highlight its different components or functionalities or characteristics such as price, durability, reliability, ease-of-use, and so forth? That requires breaking down the product—a composite of many aspects—into smaller sub-aspects and addressing these sub-aspects separately. And, to reassure potential customers of, say, the safety of the product, one might have to refer to common attributes shared by the new product and other existing, known safe products. This requires recognizing common patterns, say, in the design of the product, or quality control process, or the materials used to manufacture the product. Similarly, to make the product stand out, one might have to tease out its uniqueness away from existing products. And this also requires pattern recognition.

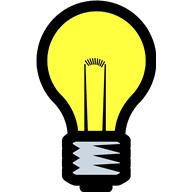
 There are many benefits to pay close attention to an object or a situation. Looking at something familiar in a new way or from a new point of view is part of Surrounding, a creative thinking skill. Paying close attention to things you ordinarily would not notice or would take for granted can be a rich source of new ideas. You may find inspiration in something you use everyday and then be able to apply that solution to another domain, a form of Generalization, a computational thinking skill. Being able to take a fresh look at a situation to see aspects or possibilities you might have overlooked is also part of the collaborative/process skills of Persisting and Playing. In art, looking at a subject without preconceptions and as if you’re seeing it for the first time is essential to developing representational drawing skills. This “beginner’s mind” also is a way of discovering unexpected uses for conventional tools.

 A “function” is an activity or purpose which is natural, ordinary or intended for a person or a thing. When we describe something as functional, we mean that it works or operates as intended. When you describe the functions of your object in this exercise, you’ll be describing its uses or the tasks it performs. A function is also a basic task of a computer. A function in computing is a kind of procedure or process and usually refers to a section of a computer program that performs a specific task, and the function can be “re-used” to perform the same specific task repeatedly at different times, and even by other programs. Note: Think about a colander that can be used again and again by different people in a kitchen. Or think about the square-root function on your calculator or smartphone.

In this sense, you could think of your object metaphorically as a computer program or a collection of instructions to perform a specific task. Your object has at least one function, with inputs (something from an outside source) and outputs (the input has been manipulated or transformed in some way). In our example of a colander, one of its functions is to drain pasta after cooking. In that case, cooked pasta would be the input and drained pasta would be the output. You could diagram this function as

Cooked pasta and cooking water → Drain → Cooked pasta without water.

You could make a similar diagram for the colander’s Steam function.

The 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 formula (an algorithm) used to mix the concrete for a road or bridge is unclear, workers may make a mistake during pouring leading to a reduced service life or even structural failure. 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.

For example, in this problem of describing an everyday object, we carry out a series of steps: (1) choosing the object that we want to describe, (2) identifying its physical characteristics, (3) identifying its functionalities, and (4) reviewing the description to retain, for example, the truly *definitive* physical characteristics and functionalities and removing others. Some of these steps can be carried out concurrently, such as having two students performing steps 2 and 3 separately. Additional steps can also be added to the process such as brainstorming and peer review of writing.

### 1. Week One [20 points]

### 1.1. Written Description on your Group Wiki Page

Over the course of the first week, generate your written description of your object. Use your group wiki page, titled Everyday Object Week 1 Description to develop your description as a group.

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 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 must have at least one function, at least one need, and a minimum of 6 physical characteristics for your object to receive full credit. Keep in mind that physical characteristics may involve all of your senses.

**Important:** **In order to receive credit for Week 1, please make sure that you add or edit material to the group wiki page so that you appear as an editor or commenter on the page.**

### 2. Week Two [20 points]

### 2.1. Analysis and Reflection on your Group Discussion Board [20 points]

Post your Analysis and Reflection responses in your Group Discussion Board, NOT on your group wiki page. Use the Everyday Object Week 2 Analysis and Everyday Object Week 2 Reflection threads for your responses.

You are expected to discuss these analysis and reflection questions among your group. In the first post of the Analysis or Reflection threads, paste in the Analysis or Reflection questions. Then, using the Analysis or Reflection questions as prompts, each member will post his or her response as a reply to the first post in the Analysis or Reflection thread.

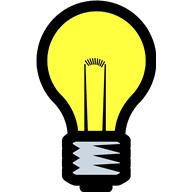
You will be graded individually based upon your contributions to the group Analysis or Reflection. In order to receive individual credit for Week 2, each group member must contribute to the answers to these questions on the Analysis and Reflection threads. **Group members who do not contribute to the Analysis or Reflection threads will not receive points.**

### 2.2. Analysis [10 points]

**Respond to these questions:**

**Analysis 1 [5 points]:** Considering your object as a computer program, identify its functions (your object has at least one function and may have more) and give each function a name. Draw a diagram that shows each of your object’s functions as a box and for each function, show the inputs and outputs. Are there any shared inputs and outputs among the functions?

**Analysis 2 [5 points]:** Looking at your list of physical characteristics, can they be further organized into a hierarchy of physical characteristics such that some of “sub-“characteristics of others? Why or why not? If yes, show examples

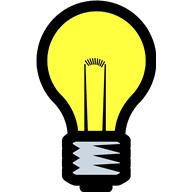
This diagraming process is important for problem analysis in computer science particularly and in all problem solving in general. Just as we have organized similar blocks of code using functions, we can organize functions with similar inputs and outputs together. This process provides a “big picture” 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.

### 2.3. Reflection [10 points]

**Respond to these questions:**

**Reflection 1** **[5 points]:** Considering your response to Analysis 1, are there functions that can be combined so that the object could 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?

**Reflection 2 [5 points]:** Do a Google image search for your object and consider the common variations of your object you encounter. How have you used abstraction to describe your object: (1) Is your description inclusive enough to cover the most common variations of your object as shown in your search? If not, what characteristics or functions would you add to your description? (2) Is your description too detailed, so that it excludes common variations of your object? If so, what characteristics or functions would you take away from your description?

 Recall that abstraction is filtering out information that is not necessary while retaining information that is useful when solving a particular problem. When we describe an object, we practice abstraction as we don’t describe every single detail. Indeed, using the computational thinking skill of abstraction, the challenge is to have enough detail that the object is recognizable but not so much detail that we exclude common examples of the object. Think about class design in object-oriented programming where we define for each class a set of members attributes and member methods (functions). For example, in Java, when we define a class, we need to decide what private variables and constants to include, and these are akin to physical characteristics of an everyday object. And we also need to decide what public methods to include, and these are akin to the functionalities of an everyday object.

Furthermore, the abstraction process is indeed 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 – [XXX, 11:59 p.m.]:** You should have completed the description of the object by the Week 1 deadline above. This description should be posted to the body of your group’s wiki page.

**Week 2 Deadline – [XXX, 11:59 p.m.]:** On the Everyday Object Week 2 Analysis and Week 2 Reflection threads on your Group Discussion Board, you should have posted your individual Analysis and Reflection comments.

# Grading

Week 1. Object description posted in the body of the group wiki page. Each group member must contribute to the description by writing or editing to receive points. The description must include at least one function, one need and six physical characteristics for full credit.

Week 2. Analysis and Reflection (on the Group Discussion Board): graded individually. Each member must post in **each** Week 2 Analysis and Reflection threads with a minimum of 1-3 coherent, relevant sentences for full credit.

**Late work will not be graded.**

### Appendix A. List of Objects

zipper

mechanical pencil

binder clip

ziploc bag

scissors

tape measure

stapler

nail clippers

umbrella

flashlight

can opener

clothespin

toilet paper holder

revolving door

pliers

ball point pen

mousetrap

screwdriver

belt

### Appendix B. Example of Patent Description for Scotch Tape

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.

END OF EXERCISE