Assignment 3: Nuclear Reactor

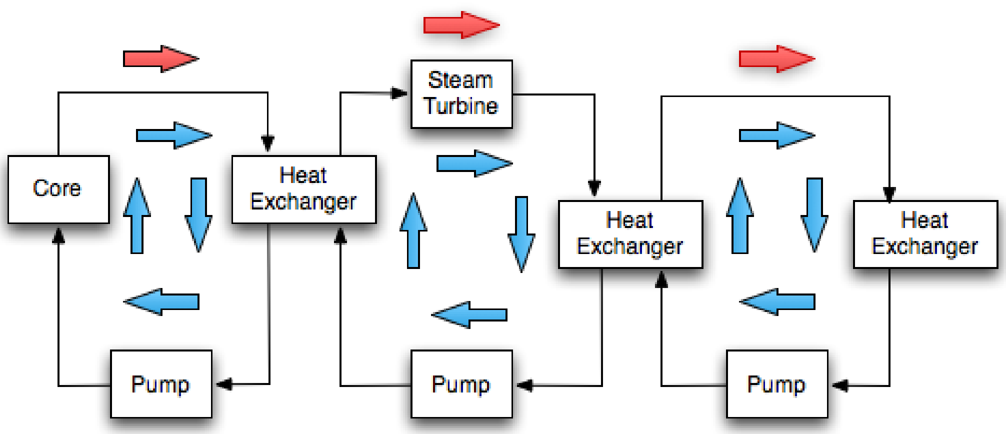
Weight: 15%	
Due: Wednesday, April 1, 2009 at 11:59pm (Analysis & Design Document)	
Due (Modified): Friday, April 17, 2009 at 11:59pm	

Assignment Goals

The purpose of this program is to give you practical experience in the analysis, design and implementation of an object-oriented system where that system contains classes which can be generalized. Similarly, you will be given a display class which you will have to integrate into your solution. In this assignment, you will be simulating a nuclear reactor. A reactor is a complex system, but it can be constructed by creating a series of small, discrete units and then assembling them into the more complex system.

Nuclear reactor basics

For the purpose of this assignment, you will be simulating a pressurized water reactor (the same kind involved in a partial core meltdown in 1979 [1]). The basic structure of the reactor is shown in the following schematic:



The reactor is constructed of three water loops: Main, Secondary and Tertiary. The main loop (which is under very high pressure to keep the water within the loop from turning into steam) extracts heat energy from the core. The core generates heat through a nuclear reaction. Energy is transferred from the main loop to the secondary loop through a heat exchanger. In the secondary loop, the pressure is not as high as the main loop so the water boils and becomes steam. The steam in the secondary loop drives a steam turbine to generate electricity. Once through the turbine, the steam must be condensed back to water so it is pumped through another heat exchanger. The second heat exchanger transfers heat to the tertiary water loop (thus condensing the steam in the second loop back to water). The water in the tertiary loop is then cooled in the cooling towers.

The blue arrows in the diagram above show the movement of water within the three water loops. The red arrows show the flow of energy through the system in the form of heat.

Assignment Specifications

Your job in this assignment is to develop a working simulation of this reactor. Your initial tasks will focus on the development of an object model of the system. You will then classify and implement the objects within that object model. Through observation, you will notice commonality between those classes which you can then abstract into a generalized class. These classes will be integrated with the Display class and interfaces provided by your instructor to produce your final product. In order to implement each kind of object, you will have to understand how they work:

- Core: The core adds energy into the system. The amount of energy is controlled by controlling the rate of the nuclear reaction using control rods embedded into the core.
- Pump: In order for energy to move through a loop, the water must be pumped. The amount of pressure determines how much energy is moved from one component to the next.
- Heat exchangers: Energy moves from one loop to the other through heat exchangers. Heat exchangers will be described in more detail later.
- Steam turbine: The steam turbine takes energy out of the system by converting it to electricity.

Heat exchangers

Heat exchangers work on the following principle: If a pipe carrying hot water is in physical contact with a pipe carrying colder water, heat will naturally conduct through the pipes from the hot water to the cold water. The temperature difference between the water in the different pipes determines how much heat energy will be transferred. For example, if pipe A contained water with a temperature of 100 degrees and pipe B contained water of 0 degrees, heat energy would transfer from the water in pipe A to the water in pipe B up until the point that the temperatures would match (in this case at 50 degrees). Therefore, there would be 50 degrees of heat energy transferred to pipe B. If instead, the temperature of the water in pipe B was initially 50 degrees, heat energy would transfer from the water in pipe A to the water in pipe B until the temperature of the water in each pipe was the same (in this case, at 75 degrees). The transfer in the second case would be 25 degrees of heat energy.

Units of Energy

In real nuclear reactors, the computation of energy stored as heat in water is a very complex mathematical process. Similarly, determining how much heat energy can be transferred through water pipes based on water pressure is also difficult to compute. Finally, computing the amount of electricity produced through a steam turbine also has its challenges. To simplify the simulation, we will simply be using a single unit of energy and we won't worry about converting between energy in different forms. Similarly, each component will have a rating based on this energy unit. These ratings are:

- Core: The core can have a production value between 0 and 100. 0 means the core is producing no energy and 100 means the core is producing maximum energy. To simulate the core adding energy to the system one simply needs to add the core's energy production to the energy contained within the system for each simulation step.
- Pumps: Pumps have a pressure value between 0 and 100. 0 means that the pump is off and 100 means the pump is operating at full. When a pump is operating at 100, it will transfer up to 100 units of energy from

one component in the loop to the next for each simulation step. If a pump is operating at 50, it will transfer up to 50 units of energy from one component to the next for each simulation step.

- Heat exchangers: Heat exchangers have two sides: the heat source and the heat sink. Energy transfers from the source to the sink based on a simple calculation:

$$\text{transfer from source to sink} = \frac{\text{energy at source - energy at sink}}{2}$$

- Cooling tower: The cooling tower is a special kind of heat exchanger. The amount of energy transferred is based on a percentage between 0 and 100. 0 means that no energy is being transferred to the atmosphere and 100 means all energy in the system is being transmitted to the atmosphere
- Steam turbine: The turbine converts energy to electricity based on its rate of efficiency. If, for example, a turbine's efficiency is 50, that means that during any simulation step 50 % of the energy fed into the turbine is removed from the system and released as electricity and 50 % remains within the secondary water loop.

Controllable parameters

In this simulation, we are able to control the following parameters:

- Rate of production of the core (0-100)
- Efficiency of Steam turbine (0-100)
- Efficiency of Cooling towers (0-100)
- Pressure at pump 1, pump 2 and pump 3 (0-100)

In real life, you cannot simply change the efficiency of the turbine and the cooling towers, but for the purpose of the simulation we can adjust the efficiency values to see what effect it would have on the simulation. Heat exchangers are not controllable; they exchange heat based on the temperature differential defined above.

Running the provided solution simulation

The instructor has provided a solution to this assignment. It is recommended that you play with the simulation enough so that you can understand how the system works. When you start the simulation, the controls are set up so that the reactor heats up quite quickly. You should let the reactor get to a good running temperature (say 200-300 units of energy) before taking action which stabilizes the temperature in the core. The reactor is a complex system. The goal of a simulation is to find a series of control settings so that the reactor stabilizes while producing a constant flow of electricity output. Similarly, you might want to explore different configurations to see if you can maximize electricity output while minimizing energy losses at the cooling towers.

Integrating with the graphical display

The instructor has provided all the necessary code for displaying the simulation. **YOU DO NOT NEED TO MODIFY THIS CODE** in order to integrate your solution with the Display. If you modify the Display class you will lose marks. In order for you to integrate your solution with the display code, your classes will have to implement to two interfaces which are provided: *ReactorInterface* and *ReactorComponentInterface*. The ReactorInterface contains the following method definitions:

```java
public interface ReactorInterface
{
    public ReactorComponentInterface get(String name);
}
```

This means that you must provide an object which allows the display system to get a reference to an object which implements the ReactorComponentInterface based on the name of the component. In your system, your components must be given the following names:

- Reactor Core
- Main Pump
- Secondary Pump
- Tertiary Pump
- Heat Exchanger 1
- Heat Sink 1
- Steam Turbine
- Heat Exchanger 2
- Heat Sink 2
- Cooling Tower

These are the names of components that the Display system expects to see. The second interface is defined as:

```java
public interface ReactorComponentInterface
{
    public List<String> getDisplayText();
    public String getName();
    public int getSimulationValue();
    public void setSimulationValue(int aValue);
}
```

Reactor components must provide the following methods:

- getDisplayText: return a list of strings which contains the text to be displayed for the component
- getName: return a string which represents the name of the component (the name must be the same as the names above: "Reactor Core", "Main Pump", etc)
- getSimulationValue: return an integer value which represents the control value for the component (0-100)
- setSimulationValue: sets the control value of the component to a specified integer (0-100)

Evaluation

Your mark for each part will be calculated as follows:

	Excellent	Satisfactory	Unsatisfactory
Documentation	(10 marks) Your documentation is effective, concise and includes all of the necessary components.	(5-9 marks) Your documentation is missing one or two necessary components, you are overly verbose in a few places, or it is difficult to understand what you have written for one or two descriptions.	(0-4 marks) Most of your documentation is missing the necessary components; your comments are extremely long and usually difficult to understand.
Programs functions correctly	(10 marks) Your program produces correct output according to the specifications.	(5-9 marks) Your program produces mostly correct output, with the exception of up to four errors.	(0-4 marks) Your program produces incorrect output in more than four cases.
Implementation	(10 marks) It is very easy to follow the flow of your program and it is clear why each step is performed and each class was included. You divide your code across several classes with methods of relatively small size.	(5-9 marks) The TA has some difficulty understanding why you chose the classes and methods that you did, but is able to eventually figure it out. Some of your methods are long and could be broken down into two or more functions.	(0-4 marks) The TA has a very difficult time understanding your implementation (or cannot understand it at all). Most of your methods are too long and could be broken down into two or more.
Class, method, and variable names	(10 marks) The names you chose make your code clear and easy to read.	(5-9 marks) Up to six names aren't clear (e.g., x, foo, bar, class A).	(0-4 marks) More than six names aren't clear.
Demonstration (Modified)	(10 marks) Your program works according to the specification during the demonstration.	(5-9 marks) Your program works according to the specification, with the exception of up to three test cases.	(0-4 marks) Your program produces incorrect output for most or all of the test cases.
Analysis and design	(25-30 marks) Your class model has a minimal number of classes and makes your code clear and concise.	(15-24 marks) You use 3-5 classes that are unnecessary, redundant, or in some way complicate your design.	(0-14 marks) In more than five places, you choose inappropriate classes.
Design Document (10 marks) Final Write-up (10 marks)	(10 marks) Your documents are clear, concise, and contain all of the stated components.	(5-9 marks) Your documents are difficult to understand or do not include up to three components.	(0-4) Your documents are missing or do not include more than three components.

(Modified) There will be no formal demonstration for this assignment. Instead, the TA will perform a set of tests on your code on his own. The TA may deduct up to 5% from the assignment's final mark for errors in spelling and grammar.

Final Write-up

Write a small report (maximum 2 pages) which describes what you learned from this assignment. Some topics you may want to include are:

- How did this assignment help to reinforce your understanding of the process of generalization?
- How did this assignment help to reinforce your understanding of polymorphism?
- How did this assignment help to reinforce your ability to analyse complex systems?
- Did you learn something about the operation of nuclear reactors? If so, what?
- When you went through the process of generalization, how much code did that remove from your solution?

Pair Programming

If you choose, you may work with a partner on this assignment. Both partners will hand in a single assignment and will receive the same mark on the assignment.

Design Document

An analysis document is due on April 1, 2009. This document should clearly show your analysis of the system. It should include an object model diagram and a class model diagram as well as any supporting documentation. It is worth half of the documentation component of the final evaluation.

Demonstration (Modified)

Your TA will execute your program to ensure that it is behaving as specified using a variety of test cases. There will be no formal demonstration of your code outside of this test and you do not need to set up a meeting with your TA for this assignment.

Handing in your assignment

For this assignment, email your program and your write-up to your TA on or before the due date. Be sure to check that the format of your final report is one which your TA can read. Be sure to include all files with the .java extension (including those provided to you that you make use of) as well as a description of how to run your program (i.e., which class contains your 'main' method). Make sure that your email client program saves a copy of the email you send to your TA. In the event of email problems, we need the header information from your original email to ensure that you submitted your assignment on time.