AERSP 424: Advanced Computer Programming

Homework 2 Spring 2024

Submission Instructions:

- Submit a .zip file containing your .cpp and/or .h files.
- Bonus points (up to 5 points) if you use Git, have a history of commits, and submit a URL to your repository.
- Your code needs to be compilable and contains some comments.
- Only the final submission will be graded.
- The submission deadline is at 11:59 PM on Friday 4/5/2024 (The late policy mentioned in the syllabus will be applied).
- Explicitly declare variables with a proper name (easy to understand) and datatype (reflect the real-world scenario *if possible*).

Question 1 (20 points):

In an aerospace control system, different types of sensors are used to monitor various parameters such as temperature, pressure, position, velocity, attitude, altitude, heading, airspeed, etc. Each sensor type has a unique way of gathering and processing its data. Your task is to design and implement a simplified sensor simulation for an aerospace control system that can handle multiple sensor types and adapt to new types in the future. The simulation must collect data from each sensor, process it according to the sensor type, and then adjust the aircraft's controls based on the processed data.

1.1) Implement an abstract base class "`Sensor`" with a virtual function "`gatherData()`" to simulate data collection and another virtual function "`processData()`" for data processing.

1.2) Pick three sensors used in an aircraft, then implement three derived classes, one for each sensor, from the base class "`Sensor`", each with its own implementation of "`gatherData()`" and "`processData()`". The implementation of "`gatherData()`" function prints out

"*Gathering data from <name_of_the_sensor> Sensor.*" The implementation of "`processData()`" function prints out "*Processing data from <name_of_the_sensor> Sensor.*"

1.3) Create another class named "`SensorFactory`" that contains a static function called "`createSensor()`", which receives a string as an input and returns a pointer to a `Sensor` class as an output. The implementation of this function checks if the input string matches with the sensor type, then allocates a heap memory for that sensor and return it. Otherwise, return a null pointer.

1.4) Implement a control system class "`AerospaceControlSystem`" that has a function "`addSensor()`", which receives a pointer to a `Sensor` class. It also has another function "`monitorAndAdjust()`" that iterates through all sensors, calls "`gatherData()`" and "`processeData()`" functions for each sensor. Then, print out "*Adjusting controls based on sensor data.*" at the end of each iteration.

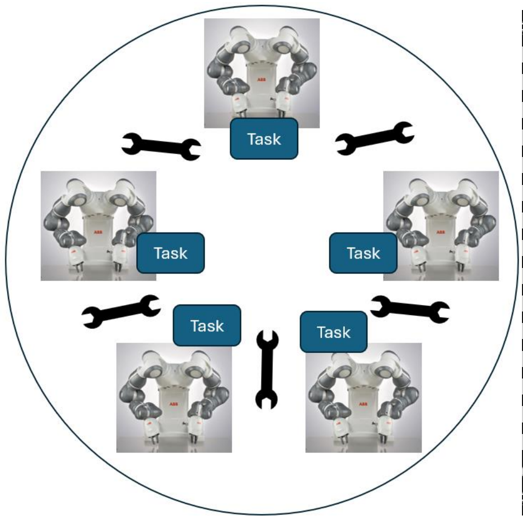
1.5) In the "`main()`" function, instantiate an object named "`ctrlSys`" from the "`AerospaceControlSystem`" class. Through that object, invoke its "`addSensor()`" function three times, each for each type of sensors. Then, invoke its "`monitorAndAdjust()`" function.

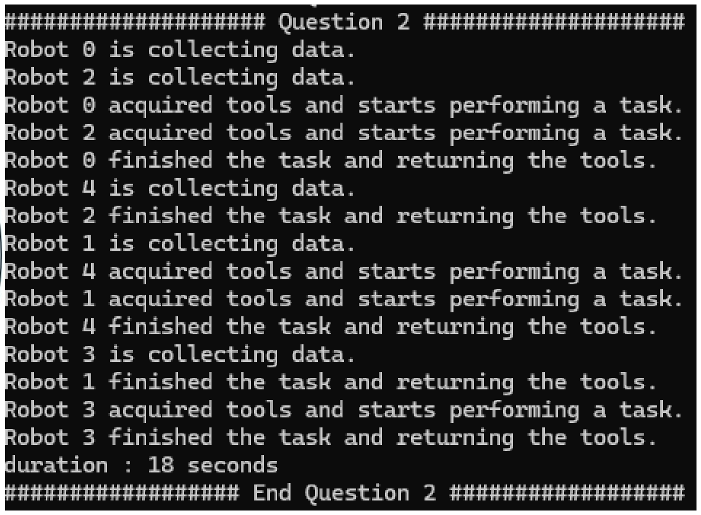
<u>Note</u>: A process in step 1.3) is commonly known as a Factory Method.

<u>Question 2</u> (30 points):

Consider a group of five robots placed together in the same circular arena as illustrated in the figure below. Each of them is assigned a task. There is one tool between a pair of robots (five tools in total). Each robot needs to grab both tools adjacent to them in order to complete the task. Every robot can perform the task simultaneously; however, they cannot start the task until they have both tools. Assuming a robot needs one second to reach and grab the tools. Then, it takes an additional five seconds to complete the task and return the tools to their original location. The robot can only perform its own task and cannot help other robots.

Write a C++ program that simulates this scenario and minimizes the total time that all robots (altogether) use to finish all tasks.





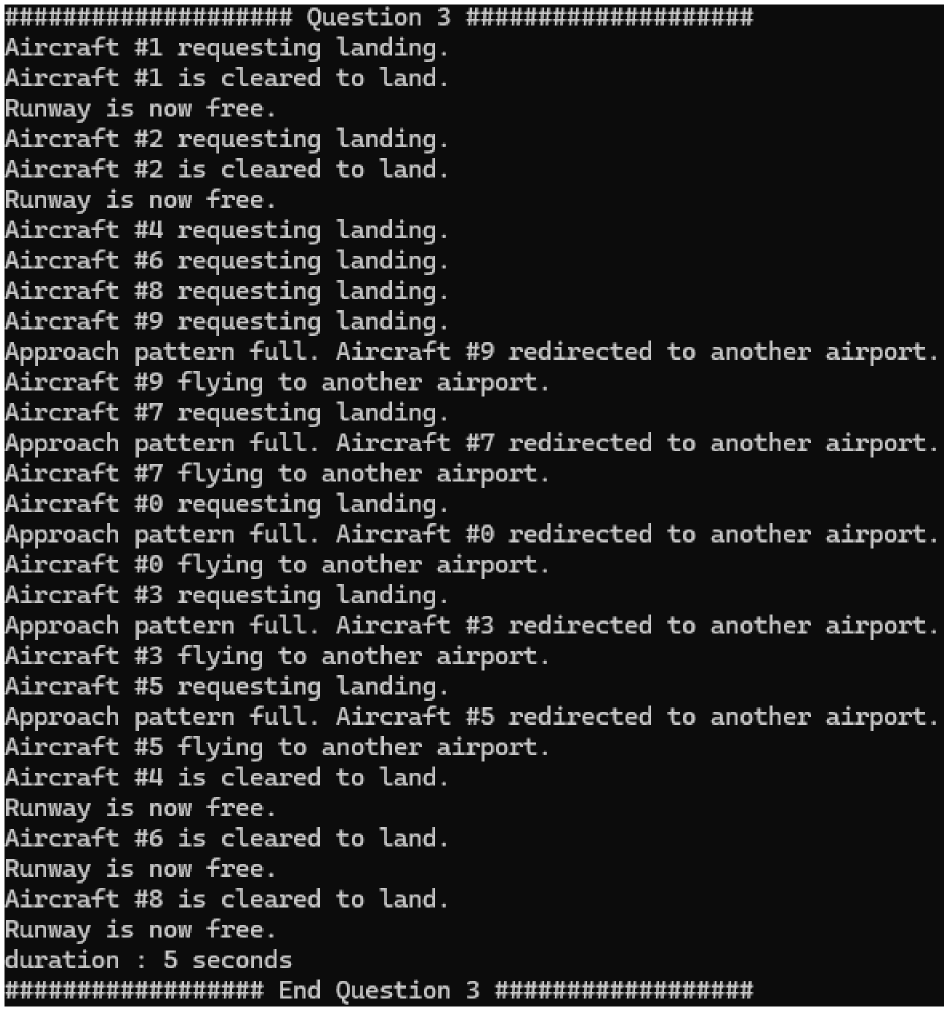
```
################### Question 2 ###################
Robot 0 is collecting data.
Robot 2 is collecting data.
Robot 0 acquired tools and starts performing a task.
Robot 2 acquired tools and starts performing a task.
Robot 0 finished the task and returning the tools.
Robot 4 is collecting data.
Robot 2 finished the task and returning the tools.
Robot 1 is collecting data.
Robot 4 acquired tools and starts performing a task.
Robot 1 acquired tools and starts performing a task.
Robot 4 finished the task and returning the tools.
Robot 3 is collecting data.
Robot 1 finished the task and returning the tools.
Robot 3 acquired tools and starts performing a task.
Robot 3 finished the task and returning the tools.
duration : 18 seconds
################# End Question 2 #################
```

<u>Question 3</u> (30 points):

Imagine a hypothetical airport with one overexploited air traffic controller personnel, one runway, and an airspace where the traffic pattern can only contain three aircraft at the maximum. Hence, the following rules apply.

- If there are no incoming aircraft, the ATC personnel fall asleep.
- When there is an incoming aircraft, the pilot needs to check how many aircraft in the traffic pattern are, and check if the ATC is talking to another pilot.
- If the ATC is not talking to another pilot (asleep), the pilot in the traffic pattern can establish a communication, which will wake the ATC up.
- If the traffic pattern is not full but the ATC is talking to another pilot, the pilot can enter the traffic pattern.
- If the traffic pattern is full, the pilot needs to divert to other airports.
- If the ATC finishes talking to a pilot, the ATC falls asleep until the next pilot initiates a communication.

Write a C++ code to simulate this scenario where ten aircraft are coming to this airport. Assume that the landing process takes one second and all other actions, previously mentioned, can happen immediately (in the order of nanoseconds, depending on how fast your computer is) without any significant time delay.



```
#################### Question 3 ####################
Aircraft #1 requesting landing.
Aircraft #1 is cleared to land.
Runway is now free.
Aircraft #2 requesting landing.
Aircraft #2 is cleared to land.
Runway is now free.
Aircraft #4 requesting landing.
Aircraft #6 requesting landing.
Aircraft #8 requesting landing.
Aircraft #9 requesting landing.
Approach pattern full. Aircraft #9 redirected to another airport.
Aircraft #9 flying to another airport.
Aircraft #7 requesting landing.
Approach pattern full. Aircraft #7 redirected to another airport.
Aircraft #7 flying to another airport.
Aircraft #0 requesting landing.
Approach pattern full. Aircraft #0 redirected to another airport.
Aircraft #0 flying to another airport.
Aircraft #3 requesting landing.
Approach pattern full. Aircraft #3 redirected to another airport.
Aircraft #3 flying to another airport.
Aircraft #5 requesting landing.
Approach pattern full. Aircraft #5 redirected to another airport.
Aircraft #5 flying to another airport.
Aircraft #4 is cleared to land.
Runway is now free.
Aircraft #6 is cleared to land.
Runway is now free.
Aircraft #8 is cleared to land.
Runway is now free.
duration : 5 seconds
################## End Question 3 ##################
```

Note: Only consider the approach to landing aircraft. Ignore taking off and taxiing.

Question 4 (10 points):

Pick your favorite equation during your aerospace engineering program, and write a C++ program to plot a 2D graph of the equation using any graphic library of your choice. You can pick any parameter for the x- and the y-axis, and initiate other parameters with a realistic value.

Question 5 (5 points):

From Question2, what issues may happen if the robot that finished the task is assigned a new task immediately? And what will be a potential solution for the issues?

Note: The code is optional for this question. You can submit the discussion as a text file.

Question 6 (5 points):

From Question3, what issues you might obtain during the development of the program? What issues do you think will happen if some of the actions, e.g., waking up the ATC, checking the traffic pattern, entering the traffic pattern, diverging to another airport, etc., take time to finish? And what will be a potential solution for the issues?

Note: The code is optional for this question. You can submit the discussion as a text file.