Assignment 2:

Object-Oriented Analysis and Design

Name:

# Part 1: Identifying Objects

Highlight **all** the nouns (i.e., potential objects) in the following program description:

Create a program that simulates the growth of two plants. For each plant, the user provides a name, starting height, and daily growth rate. The user is then provided a choice with 5 options: (1) display plant information; (2) compare the heights of the plants over a specified number of days; (3) determine on what day the height of the fastest-growing plant will be greater than or equal to the height of the slowest growing plant; (4) change plant information; (5) quit the program.

# Part 2: Identifying Object Datatypes

Complete the following table using the objects identified in Part 1. Extend the table’s rows as needed.

If the object should be a data type that needs to be defined in a custom class, propose a class name in the second column. If already defined, specify which type the object is in the third column.

|  |  |  |
| --- | --- | --- |
| Noun/object | Datatype (Custom Class) | Datatype (Pre-defined) |
| Program |  | Program |
| growth |  | float |
| name |  | str |
| Starting height |  | float |
| Daily growth rate |  | float |
| User |  | str |
| height |  | float |
| rate |  | float |
| choice |  | int |
| options |  |  |
| information |  | str |
| heights |  | float |
| days |  | int |
| Fastest growing plant | Plant |  |
| Slowest growing plant | Plant |  |
| plant | Plant |  |
| plants | Plant | list |

# Part 3: Class Attributes and Methods

Complete the following table using the datatypes identified in Part 2. Extend the table’s rows as needed. Each class should have a row, with the names of each attribute and method specified within the row.

Revisit and refine this table, if required, when the program has been completed. Use this table to help construct the required UML.

|  |  |  |
| --- | --- | --- |
| Class | Attributes | Methods |
| Plant | Name  Height  growthRate | grow(self, daysL int)  displayInfo(self)  compareHeight(self, otherPlant, days) |
| Program | plants  options | displayMenu(self)  executeChoice(self, choice)  comparePlants(self, days)  findDayGrowthExceeds(self)  updatePlantInfo(self)  quit(self) |
|  |  |  |
|  |  |  |
|  |  |  |