*2022-2023*

**OSU Supplemental Instruction Session Planning Form**

SI Leader: \_\_\_\_Thomas Morton\_\_\_\_\_\_\_\_\_ Week of Semester:\_\_\_\_\_\_\_\_\_\_ 14\_\_\_\_\_\_\_\_\_\_

Course: \_\_\_\_\_\_CS 1113\_\_\_\_\_\_\_\_\_\_\_ Instructor: \_\_\_\_\_\_\_\_Dr. Crick\_\_\_\_\_\_\_\_\_\_\_\_

Session Objectives

1. Students will understand how to define objects by conceptualizing real-world scenarios

2. Students will understand the difference between static and non-static methods

3. Students will understand object instantiation by coding a small racing game using car objects

Professor Meeting Notes:

Dr. Crick mentioned that I should focus on explaining how objects act as a collection of like data and functions that act upon that data. Emphasis should be placed on storing data over manipulating data, but simple object methods are fine to discuss.

I was also asked to encourage participants to take the practice exam, and that its format will strongly resemble the final exam.

Opening/Introductory Activity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  K W L | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  5 minutes | Materials Needed:  None | Targeted Learning Style(s):  Audial | Bloom’s Levels Used:  Understanding  Remembering |
| Explanation/Notes:  Participants will be asked to describe what they know and want to know about objects & classes. This activity will be used to gauge participant understanding of the material thus far and serve as a jumping point to material that needs reviewed first.  To spur discussion over the concepts, the SI leader will ask questions about methods, classes, objects, and static vs non-static method declaration. | | | | | |

Main Session Activity 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  That’s the  Object-ive | Session Objective(s) Met:  Objectives 1 & 2 | Time Allotted:  10-15 minutes | Materials Needed:  Chalk/board  Marker/whiteboard | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Going around the room, participants will be asked to think of any object of their choosing and draw it on the board. After 3-5 objects have been drawn, participants will be asked to think of the data members and methods that may be associated with the object and write them underneath the object’s picture. Methods may or may not be parameterized; this depends on the object.  While defining the methods, participants will be asked if the method in question is static or non-static and then asked to explain their reasoning. Other participants will be encouraged to give their input on these definitions. | | | | | |

Main Session Activity 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  The Random Race | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  20-25 minutes | Materials Needed:  Computer with Java compiler if enough participants,  Otherwise, SI leader walkthrough & provided solution | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Creating  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Participants will be split into groups of 2-3 to write a simple program that “races” two modes of transport. For simplicity, these objects do not have to be instantiated from the same class definition as object instantiation will be covered in detail at the end of the semester. Objectives are as follows:   1. Write a main method that creates an object of type car and another object of type TurboDonkey. 2. Create a random object 3. Write a while loop that runs indefinitely and contains an if block performing the following:   3a. If car.drive(random.nextInt() % 2) == 0, print “Car Won!” and break;  4b. If TurboDonkey.drive(random.nextInt() % 2) == 0, print “TurboDonkey Won!” and break;  4c. Else, continue   1. In a separate file, define a class to contain a car or truck object 2. Create a random object 3. Give the object an integer data member called milesPerHour with a value random.nextInt() % 10 + 10 4. Give the object an integer data member called lapsRemaining with a value of 3 5. Give the object an integer data member called milesTilNextLap with a value of 30 6. Define a method called drive() that performs the following:   9a. takes an integer called turbo as a parameter  9b. subtracts (milesPerHour + turbo) from milesTilNextLap  9c. If milesTilNextLap <= 0, adds 30 to milesTilNextLap and decreases lapsRemaining by one  9d. returns lapsRemaining   1. Copy the above code from 4 – 9 to another file for another class called TurboDonkey with the following modifications:   10a. TurboDonkey has a speed of random.nextInt() % 7 + 10  10b. TurboDonkey adds 1 to the turbo parameter if the turbo is >= 1 before performing step 9b.    This looks lengthy, but steps 6-8 are just 3 lines of code and step 10 just copy/pastes steps 4-9 and takes one edit and one additional if statement.  (It took 35 minutes for me to come up with the activity, write this outline, write an outline for the code walkthrough, write a solution for the code, then troubleshoot the solution to make sure it is functional)  This activity reinforces writing and using classes between files and also walks through the concepts of defining data members and object methods and calling these methods from main() to update the data members of the object. | | | | | |

Closing Activity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  K W L  (Recap) | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  5 minutes | Materials Needed:  None | Targeted Learning Style(s):  Audial  Visual | Bloom’s Levels Used:  Analyzing  Understanding  Remembering |
| Explanation/Notes:  Participants will be asked to describe what they know, want to know, or have learned from the session today. Additional want-to-knows will be redirected as necessary or explained more thoroughly. | | | | | |

|  |  |
| --- | --- |
| Plan for extra activity:  If there is time remaining, the main activity 2 will be revisited and the similarities between Car and TurboDonkey will be discussed. Participants will be asked to describe whether they think it is possible to reduce the overlap in code between the two.  This will provide an excellent primer to object instantiation in the last week of semester. | Extra notes: |