*2022-2023*

**OSU Supplemental Instruction Session Planning Form**

SI Leader: \_\_\_\_\_\_Thomas Morton\_\_\_\_\_\_\_ Week of Semester:\_\_\_\_\_\_\_\_\_12\_\_\_\_\_\_\_\_\_\_\_

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

Session Objectives

1. Students will understand variable scope with respect to user-defined methods

2. Students will understand how methods provide abstraction in code, resulting in easier debugging and code reusability.

3. Students will apply concepts of user-defined methods to write a simple program that prints data types using method overloading.

Professor Meeting Notes:

Dr. Crick and I discussed how abstraction provided by methods is an essential concept for students to understand. With this, Dr. Crick suggested that I cover how we can use methods to break a problem down into smaller components and solve those components individually.

Following that, he suggested that I cover how to properly use methods, such as passing and returning input and designing methods with reusability in mind.

Opening/Introductory Activity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Variable Scope – A Review | Session Objective(s) Met:  Objective 1 | Time Allotted:  5 minutes | Materials Needed:  None | Targeted Learning Style(s):  Audial | Bloom’s Levels Used:  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Going around the room, each participant will list one activity or instance where the state of an object or value would and would not be remembered. Examples could be baking cookies (the cookies you baked last are not the same as the next cookies you will bake, thus scope is not preserved. Versus the oven settings, which will stay turned off until the next time you turn it on).  A brief note of how methods do not retain variable scope will be discussed following the activity. | | | | | |

Main Session Activity 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Conceptualizing Methods | Session Objective(s) Met:  Objectives 1 and 2 | Time Allotted:  15-20 minutes  (aim for 20) | Materials Needed:  Marker & whiteboard | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Creating  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Each participant will be asked to think a series of actions from the following categories: [driving a car, cooking a hamburger, writing a book, playing video games, getting ready in the morning]. As participants think of the different actions required to perform these activities, they will be asked to write each action on the whiteboard under the listed category.  After each category has a thorough list of actions, participants will then be asked to think of the inputs required to carry the action through. These inputs will be listed next to each action as a parameter of the “method.”  Following this, each participant will be asked to conceptualize what the outcome of the listed action will be. This outcome will become the return statement for the given action.  A brief explanation will be given that writing a method follows this procedure (conceptualize what needs done, what is required to do it, and what the outcome of doing it will be). | | | | | |

Main Session Activity 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Coding Activity –  A Method of Many | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  15-20 minutes  (aim for 15) | Materials Needed:  Computer with internet access or java compiler  or  Pencil & paper | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Creating  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Participants will group into pairs/groups of 3s to write a simple program using method overloading. The objectives are as follows:   1. Write the program class & main method structure 2. Declare and initialize a variable of each type with data of your choice: char, int, double, boolean, String 3. Write a for loop to iterate 5 times 4. Within the for loop, write an if statement using the for loop sentinel as a parameter to call an overloaded method printInput() for each number 5. Write an overloaded method for each data type from #2 that contains a print statement that outputs “Input type: [DATATYPE] \n Value: [ARGVALUE] 6. Write an overloaded method printInput() that multiplies the int and double from #2 and returns the result 7. Outside the for loop & if statement, write a print statement calling the overloaded method printInput from #6 and prints its return value to 3 decimal places.   This activity will cover for loops, if statements, method calls, method overloading, and reinforce the concepts of arguments, parameters, and return statements.  If there are not enough participants, the SI Leader will ask participants what is required for each step and walk through the coding process for this activity. The last ~3 minutes will involve covering the SI Leader’s solution for the activity and answering any additional questions. | | | | | |

Closing Activity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Vocabulary Review | Session Objective(s) Met:  Objectives 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 quizzed over the terms used to write methods in java. The goal is to reinforce an understanding of what these words mean and to reduce headache when moving forward to object-oriented methods.  Words to review: public, static, return statement, argument, parameter, method call, method overloading, method signature, encapsulation, abstraction | | | | | |

|  |  |
| --- | --- |
| Plan for extra activity:  Identify the “Big Idea”  Participants will be asked to summarize how and why methods are useful in programming. In addition, there will be a brief discussion on how and why method overloading is useful. | Extra notes: |