*2022-2023*

**OSU Supplemental Instruction Session Planning Form**

SI Leader: \_\_\_\_\_\_Thomas Morton\_\_\_\_\_\_\_\_\_ Week of Semester:\_\_\_\_\_\_\_\_\_15\_\_\_\_\_\_\_\_\_\_

Course: \_\_\_\_\_\_\_CS1113\_\_\_\_\_\_\_\_\_\_\_\_ Instructor: \_\_\_\_\_\_\_\_\_Dr. Crick\_\_\_\_\_\_\_\_\_\_

Session Objectives

1. Students will review essential Java concepts covered over the last 15 weeks

2. Students will successfully understand when to apply different programming concepts. ie. Types of loops, branching statements, and program breakdown into objects and methods

3. Students will apply concepts of this course to review and optimize previous session programming problems. New concepts (especially methods and objects) will be applied to these problems to create more efficient, succinct, and useful programs.

Professor Meeting Notes:

Opening/Introductory Activity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  K W L | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  15 minutes | Materials Needed:  None | Targeted Learning Style(s):  Audial | Bloom’s Levels Used:  Understanding  Remembering |
| Explanation/Notes:  Participants will be asked to list one thing they know or want to know of the following topics:  primitive data types,  logical operators,  assignment operators,  branching statements (including switch),  looping statements,  ternary operators,  class and main method definitions,  method definitions,  method handling & overloading,  and object instantiation.  Want to know will be written on the board to help guide the SI leader to focus future discussion during activities towards these topics. | | | | | |

Main Session Activity 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Random Pennies! | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  20 minutes | Materials Needed:  SI Solution | Targeted Learning Style(s):  Audial,  Visual,  Kinesthetic | Bloom’s Levels Used:  Creating  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  This activity revisits the coding exercises from weeks 5 and 6. The goal of this activity will be to create a functional program that does the following:   1. Defines a main method that calls a generateRandomPennies() function    1. This function will accept no arguments and will return an integer    2. It will define a random object, which will prompt the user for a maximum number to generate (or else it can go to 2.17 billion)    3. The random object will generate this integer, tell the user its value and return it to the calling method 2. Define another method called makeChange() inside main that uses branching statements and the modulo operator to tell the user the number broken down into the least number of pennies, nickels, dimes, quarters, and dollars    1. This method will take the int from generateRandomPennies() as an argument   Afterwards, participants will be asked to think of whether this program could be rewritten to accept other data types, then give their reasoning. | | | | | |

Main Session Activity 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Jeopardy | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  40 minutes | Materials Needed:  Jeopardy questions  Candy Prizes? | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Participants will play a game of jeopardy! Questions will cover all material covered in lectures thus far, with the first round covering the first 8 weeks and the final round covering the last 8 weeks. The top two scoring contestants will receive a candy prize, and all other participants will receive a consolation prize of a crisp high-five. | | | | | |

Main Session Activity 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  A Completed Calculator | Session Objective(s) Met:  Objectives 1, 2 & 3 | Time Allotted:  35 minutes | Materials Needed:  Access to the internet to view the java.lang.Math API | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Creating  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  This activity will revisit the calculator from week 13, but this time implementing some small changes. These changes include:   1. The program will use additional math functions from the java.lang.Math library    1. New functions will include sin(), cos() and tan()    2. The calculator will also include a recursive method for factorials (This should take some time to implement) | | | | | |

Main Session Activity 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Predict Test Questions | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  20 minutes | Materials Needed:  Candy Prizes | Targeted Learning Style(s):  Audial  Visual  Kinesthetic | Bloom’s Levels Used:  Creating  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  This is a competitive twist on predict test questions. The winner will get a candy prize! Going around the room, each participant will be asked to think of a potential test question or scenario they may be asked to evaluate. Other participants will be given the opportunity to answer the question and win one point. The highest number of points by the end of the activity wins! | | | | | |

Main Session Activity 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  The Random Race Revisited | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  40 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.  Changes to this activity will be to implement parameterized constructors for both vehicle methods:   1. Ability to set milesPerHour using an int as an argument 2. Using an array as an argument, the user will be able to set the following:    1. Ability to set lapsRemaining    2. Ability to set milesTilNextLap 3. The main method will prompt the user if they want to set up their own race conditions, if so, then the participant will implement the following:    1. Create an array to store the lapsRemaining and milesTilNextLap    2. Prompt the user for the Car’s milesPerHour and then instantiate a car object    3. Prompt the user for the TurboDonkey’s milesPerHour and then instantiate a TurboDonkey object    4. Otherwise, run the program using the above randomized conditions   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. The additional objectives discuss the final topic of Dr. Crick’s lectures, which is object instantiation. This activity will likely be difficult with these new objectives, so the SI Leader will guide participants to an understanding on using the parameterized constructor to set object values (the array part should be especially difficult) | | | | | |

Closing Activity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Name:  Predict the Next Lecture Topic  &  K W L | Session Objective(s) Met:  Objectives 1, 2, & 3 | Time Allotted:  10 minutes | Materials Needed:  None | Targeted Learning Style(s):  Audial | Bloom’s Levels Used:  Evaluating  Analyzing  Applying  Understanding  Remembering |
| Explanation/Notes:  Participants will assess concepts they have learned from CS 1113. This will provide an opportunity for participants to cover concepts they still would like to know about CS 1113, and to discuss what they have learned from the session today.  Discussion will then be pivoted to cover how these concepts tie into the functionality that users expect from computer programs today. From there, participants will be asked to discuss what CS 2133 may cover and what they look forward to learning most. (It covers file input/output, graphical user interfaces, searching and sorting algorithms, space/time complexity of algorithms, objects and classes in detail, and simple data structures) | | | | | |

|  |  |
| --- | --- |
| Plan for extra activity:  Each of these activities will likely run over their anticipated time except for the predict test questions activity. That said, the extra activity is:  Identify the “Big Idea”  Participants will cover each subject from the list in the session intro and discuss how and when each of these programming concepts see use and why. Further discussion will be encouraged over whether any of these concepts are interchangeable and the reasons why or why not. | Extra notes: |