Instructions:

1. Click on File -> Make a copy

Include your IPO Chart with your code in Eclipse IDE: We will discuss and demonstrate this in the computer lab.

IPO CHART

Program name:	**Necklace Problem (Necklace)**	
INPUT	**PROCESS**	**OUTPUT**
Hint: What will the user input?	Hint: What is the program going to do with the input information?	Hint: What will the screen display after user input?
The user will be prompted to enter 2 single digit numbers (*int)* Eg. Enter the first starting number (0-9): 4 Enter the second starting number (0-9): 6	- **Sequence Storage:** A (*String*) variable, (*sequence*), is initialized with the starting numbers (*startNum1* and *startNum2*) to build the final output. - **Step Counter:** An (*int*) variable, (*steps*), is initialized to *0* to count the number of new digits generated. - **Loop Control:** A (*do-while*) loop executes repeatedly until the current pair (*currentNum1* and *currentNum2*) exactly matches the starting pair (*startNum1* and *startNum2*), signifying the end of the necklace cycle. - **Calculation:** Inside the loop, the (*int*) variable (*nextNum*) is calculated using the formula:	- **Display Sequence:** The final contents of the (*String*) variable (*sequence*) are displayed to the user, showing the entire numerical necklace. - **Display Steps:** The final value of the (*int*) variable (*steps*) is displayed, indicating the length of the cycle (the number of new numbers generated). Eg. Generating the necklace sequence... --- Sequence: 5 6 1 7 8 5 3 8 1 9 0 9 9 8 7 5 2 7 9 6 5 1 6 7 3 0 3 3 6 9 5 4 9 3 2 5 7 2 9 1 0 1 1 2 3 5 8 3 1 4 5 9 4 3 7 0 7 7 4 1 5 6 Steps taken to close the necklace: 60

	(*currentNum1 + currentNum2*) *% 10*. - **Sequence Update:** The value of (*nextNum*) is concatenated onto the (*String*) variable sequence. - **Iteration Update:** The variables are shifted for the next step: (*currentNum1*) takes the old value of (*currentNum2*), and (*currentNum2*) takes the value of (*nextNum*). - **Step Counting:** The (*int*) variable (*steps*) is incremented.	---