

IT and Me Works

Lesson Plan for Programming Strand

Topic: Structured Programming
Teacher: Karen Olmstead

Subject: IT & Me Works
Grade: 9th

Time: 1 hour

Objectives: At the end of this lesson students will:

- Understand the importance of meaning what you say.
- Understand garbage-in garbage-out integrated with sequential thinking. (Garbage-in garbage-out means that inaccurate or incorrect steps (garbage) will result in output that is not useful.)

Standards:

- Students will recognize patterns and describe and represent relations and functions.
- Students will demonstrate the interest and ability to speak purposefully and articulately, as well as listen and view attentively and critically.

Setting: Classroom.

Materials: None

Note –This activity can be used in introductory programming using any programming language. It should take place before any programming is started.

Teacher's Role:

Direct the students to write a very detailed and step-by-step list describing the steps that must be followed to create a peanut butter and jelly sandwich. A suggested approach is follow STAIR steps for building the sandwich. See handout at end of plan for STAIR steps.

Students will read their instructions to the instructor, or another student, who will follow the steps exactly. For example, if the students say to put the peanut butter on the bread, set the jar of peanut butter on top of the bread. **This can get rather messy so make sure that the surface is covered before following the instructions.**

This lesson plan can also be done using sharpening a pencil or some other activity that consist of a series of steps that must be followed to accomplish the task.

School to Career Connection: Have a programmer visit the class and discuss how important communication skills are when programming.

Employability: Communication skills, critical thinking

STAIR Statements for Building Peanut Butter and Jelly Sandwich

- **S: State the Problem-** State the problem in 1 or 2 short sentences. Use plain, simple English. Be sure that you are stating a single, specific problem. Defining the problem is VERY important. If you are building a peanut butter and banana sandwich then that is your problem, not just a peanut butter sandwich.
- **T: Tools-** List ALL of the tools you MIGHT ever use, separated by commas in paragraph form. If you list EVERY tool you MIGHT use, you will not have to revise all the way back to your tool list if the algorithm doesn't work.
- **A: Algorithm-** YOUR ALGORITHM MUST BE NUMBERED and you must put a return after each enumeration. This is the PLAN told in a step-by-step fashion. IT could also be called a recipe for fixing the problem. What are your options? What can you do? Think of all of the possibilities, then choose and describe a course of action. Be complete in your plan. Do not skip important steps or information. When you have completed your algorithm, ask yourself three questions: Does my algorithm tell HOW? Does my algorithm tell HOW? Does my algorithm tell HOW?
- **I: Implementation-** Now is where you get to actually DO something. Following your well-planned algorithm, and write the code. Tell us how it goes. You follow your algorithm, and it works or it doesn't. If it doesn't, you must revise/refine your STAIR.
- **R: Refinement-** What went wrong? How far back in the STAIR statement did you need to revise?... back to Problem... back to Tools? ... back to Algorithm? What did you change to solve your problem? Did it work the second time? ...the third time? How many revisions did you make? Write what you did down as you do it, so you will remember. If you do it first then write it down, be sure to remember as much detail as possible and just tell the story as it happened. You can then tell how you would fine-tune the program if it applies.