

MEMBER MANUAL

USING PROGRAMMING IN 4-H

INTRODUCTION

By enrolling in this project, you have accepted the challenge to develop your own guidelines for this phase of your computer learning. Before starting work on your project, read this project guide thoroughly. Familiarize yourself with the recommended steps to avoid wasting time and effort.

Through this project you will learn to apply/use various programming skills, improve your decision-making ability, develop leadership skills through group activities, and express yourself effectively through oral presentations and exhibits.

REVIEW

Review these basic programming terms before determining an "interest" for your project:

- String Variable
- Looping
- Unconditional Branching
- Conditional Branching
- Subroutines
- Arrays

Microcomputers may vary greatly from one brand or model to the next. However major limitations in programming are based on user knowledge of computer languages and techniques.

Your computer can perform math, store information, and recall data hundreds of times in less than a second without making a mistake. Yet, even with these capabilities, you will receive accurate

results only if you supply correct programming instructions to your computer.

DETERMINING YOUR INTERESTS

Selecting an interesting and worthwhile programming project will give you an opportunity to make some important decisions.

First make a list of your interests. Don't feel compelled to solve a complex problem. Start with something that excites you and gains your attention or curiosity on a more basic level.

List five to nine ideas which are of interest to you. Leave two blank lines after each idea that you have listed. Then indicate programming techniques you would use to develop each idea on the lines you left blank earlier.

Ranking Your Interests

Narrow your list to three interests you would like to work on this year. Since choosing is often the most difficult part of decision-making, consider:

- Your experiences
- The cost
- Where information is available
- What you will learn
- Whether it will help other people
- How to get additional help, if needed
- Whether you have adequate time to complete the project

On the sheet of paper where you listed your interests, circle the number in front of your top three

choices. Now pick one of these three topics to develop as your programming project for this year. Mark it on your paper with a star. At the end of your list, describe why you chose this interest and what you considered in making the decision.

Keep your list in a safe place. After completing the first project, you may have time to begin work on your second choice!

PLANNING YOUR PROJECT

Everyone plans. Plans, ranging from individual thoughts to detailed, written procedures, help you arrange future activities. Written plans are encouraged in this project to provide a "road map" of what you wish to accomplish. Without such a plan you may find yourself "lost".

On another piece of paper, develop a written plan for your project. Include your goals, how and why you chose this project, the procedures you will follow, your responsibilities, and the learning experiences you expect to gain.

Setting Goals

Goals can be compared to mileposts along a highway. They help you determine how much progress you have made toward getting where you want to be. Goal statements should indicate the growth/change that is to take place. Growth/change may take place in the following areas:

1. Feelings
2. Values
3. Beliefs
4. Skills
5. Attitudes

Developing a Plan of Action

Your plan of action should include a step by step listing of what you are going to do to accomplish your goals. The following questions may help you develop your plan of action:

1. What resources/materials will I need?
2. What additional help will be needed?
3. What basic skills do I need?
4. What learning experiences will help me meet my goals?

Once you have completed your plan of action you are ready to work on your project.

DEVELOPING YOUR PROJECT

Some of the following may help you develop and carry out your plan:

- Review the 4-H Computer Advancement Program
- Review your plans with others
- Write for literature/materials
- Review 4-H computer activity sheets
- Obtain information from magazines, newspapers, etc.
- Ask to be included in user groups
- Check libraries for information
- Visit computer stores

List your ideas for developing your project. Include the types of resources, their location, and the information you hope to gain.

EVALUATING YOUR PROJECT

True enjoyment comes with improvement or progress. Evaluate and record your progress throughout this programming project. Measure against the goals you set at the start of the project, not against the performance of another individual.

Recording Progress

Here are some ideas for recording process:

1. Take pictures and/or keep printouts of the work that you have completed.
2. Keep a diary or notebook. Include a copy of your project plan.
3. Visit knowledgeable people. Gain their insight by discussing your project with them. Keep a record of your visits, noting what you learned.
4. Enter your project in the county fair.

Sharing What You Learn

Satisfaction is often achieved by sharing what you have learned with others. The process of sharing also helps you summarize and analyze your accomplishments.

Consider sharing through:

- Meetings, workshops
- Exhibits, bulletin boards, fair booths
- Interviews, talks, radio programs
- Newspapers, magazines
- Demonstrations, illustrated talks

Members are encouraged to develop an activity guide for the project they have completed. Give a copy of your activity guide to your county Extension agent to share with 4-H computer project members and leaders.

Abridged Computer Glossary*

Applications Software. Software programs that perform a specific user-oriented task such as ration balancing or payroll. Application software can be either purchased as a package or custom-designed by a programmer.

Assembler. A computer program that converts or translates assembly language into a form, machine language, that the computer can understand. The assembler translates mnemonic instruction codes into binary numbers, replaces names with their binary equivalents, and assigns locations in memory to data and instructions.

BASIC (Beginner's All Purpose Symbolic Instruction Code). A relatively easy-to-use programming language that comes with many small computer systems.

Binary. A basis for calculations in all computers, this two digit number system consists of digits 0 and 1 which are represented in the computer as the presence or absence of small electrical pulses.

Compiler. A translation program which converts high-level instructions into a set of binary instructions for execution. Each high-level language requires a compiler or an interpreter. A compiler translates the complete program which is then ex-

ecuted. Every program change requires a complete recompilation.

Debug. To find mistakes or problems with software or hardware and eliminate them.

High-Level Language. Programming Language. Programming language in which the statements represent procedures rather than single machine instructions. FORTRAN, COBOL and BASIC are three common high-level languages. A high-level language requires a compiler or interpreter.

Interpreter. A translation program used to execute statements expressed in high-level language. An interpreter translates each such statement and executes it immediately. Instructions can be freely added or modified in the user program, and execution may be resumed without delay; an interpreter is interactive. Compare this with the compiler.

Machine Language. Set of binary codes (0's and 1's) representing the instructions which can be directly executed by a processor.

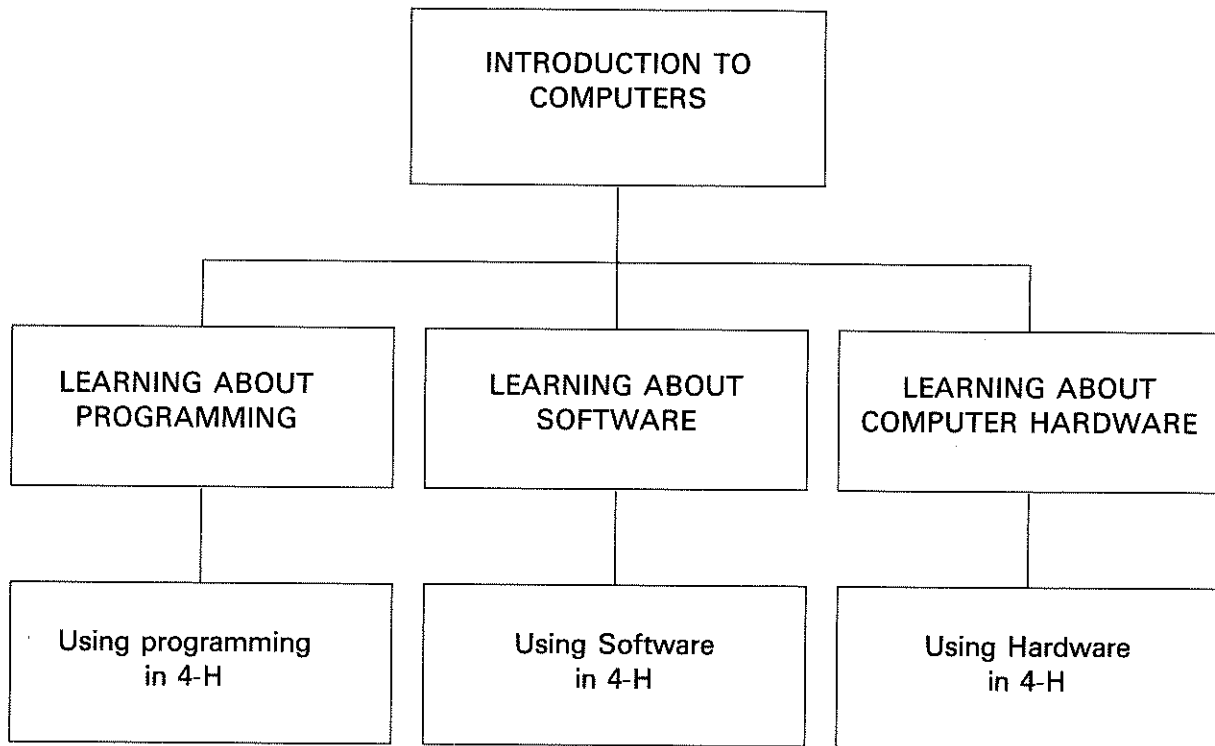
Program. Sequence of instructions directing a computer to perform a particular function; a statement of an algorithm in a programming language.

Programming Language. Set of words and rules that constitutes a language understood by the computer and program alike. See also High-Level Language.

Source Code. The humanly-readable computer commands written in a programming language. It requires an interpreter or compiler. It is sometimes referred to as a source program.

Storage. Term for any device that is capable of holding data which will be retrieved later.

*Taken from "Abridged NCCI Computer Glossary", *North Central Computer Institute Quarterly*, September 1982, pp. 5-8.



**A Pacific Northwest Cooperative Extension Publication
Washington Oregon Idaho**

This publication was written by Debra S. Marple, Washington State University, Spokane County Extension Agent, 4-H and Youth Development and James Wilson, University of Idaho, Kootenai County Extension Agricultural Agent. Sources for portions of the text included the Washington Western Games 4-H Leader Guide and "Understanding Boys and Girls," 4-H Youth Programs, University of Missouri.

Pacific Northwest Cooperative Extension bulletins are joint publications of the three Pacific Northwest states—Washington, Oregon, and Idaho. Similar crops, climate, and topography create a natural geographic unit that crosses state lines. Since 1949, the PNW program has published over 250 titles. Joint writing, editing, and production has prevented duplication of effort, broadened the availability of faculty specialists, and substantially reduced costs for the participating states.

Issued by Washington State Cooperative Extension, J.C. Engibous, Interim Director, and the U.S. Department of Agriculture in furtherance of the Acts of May 8 and June 30, 1914. Cooperative Extension programs and policies are consistent with federal and state laws and regulations on nondiscrimination regarding race, color, religion, national origin, sex, age, or handicap. Trade names have been used to simplify information; no endorsement is intended. Published July 1987. 0.25/0.25/0.25