



4.1: Steps in Program Development

IT1406 - Introduction to Programming

Level I - Semester 1

4.1. Steps in Program Development

- Computer programming is an art.
- Many people believe that a programmer must be good at mathematics, have a memory for figures and technical information, and be prepared to spend many hours sitting at a computer, typing programs.
- However, given the right tools and steps to follow, anyone can write well-designed programs. It is a task worth doing, as it is both stimulating and fulfilling.

4.1. Steps in Program Development

- Programming can be defined as the development of a solution to an identified problem, and the setting up of a related series of instructions that, when directed through computer hardware, will produce the desired results.
- It is the first part of this definition that satisfies the programmer's creative needs; that is, to design a solution to an identified problem.
- Yet this step is so often overlooked. Leaping straight into the coding phase without first designing a proper solution usually results in a program that contains many errors.
- Often the programmer then needs to spend a significant amount of time finding these errors and correcting them.
- A more experienced programmer will design a solution to the program first, desk check this solution, and then code the program in a chosen programming language.

4.1. Steps in Program Development

- There are seven basic steps in the development of a program, as follows.
 1. **Define the problem**
 2. **Outline the solution**
 3. **Develop the outline into an algorithm**
 4. **Test the algorithm for correctness**
 5. **Code the algorithm into a specific programming language**
 6. **Run the program on the computer**
 7. **Document and maintain the program**

4.1. Steps in Program Development

1. Define the problem

- This step involves carefully reading and rereading the problem until you
- Understand completely what is required. To help with this initial analysis, the
- Problem should be divided into three separate components:
 - the inputs
 - the outputs
 - the processing steps to produce the required outputs.

4.1. Steps in Program Development

2. Outline the solution

- Once the problem has been defined, you may decide to break it down into smaller tasks or steps, and establish a solution outline.
- This initial outline is usually a rough draft of the solution and may include:
 - the major processing steps involved
 - the major subtasks (if any)
 - the user interface (if any)
 - the major control structures (e.g. repetition loops)
 - the major variables and record structures
 - the mainline logic.

4.1. Steps in Program Development

3. Develop the outline into an algorithm

- The solution outline developed in Step 2 is then expanded into an algorithm:
 - a set of precise steps that describe exactly the tasks to be performed and the order in which they are to be carried out. One can use pseudocode (a form of structured English) to represent the solution algorithm.

4.1. Steps in Program Development

4. Test the algorithm for correctness

- This step is one of the most important in the development of a program, and yet it is the step most often bypassed.
- The main purpose of desk checking the algorithm is to identify major logic errors early, so that they may be easily corrected.
- Test data needs to be walked through each step in the algorithm to check that the instructions described in the algorithm will actually do what they are supposed to.
- The programmer 'walks' through the logic of the algorithm, exactly as a computer would, keeping track of all major variables on a sheet of paper.

4.1. Steps in Program Development

5. Code the algorithm into a specific programming language

- Only after all design considerations in the previous four steps have been met should you actually start to code the program into your chosen programming language.

4.1. Steps in Program Development

6. Run the program on the computer

- This step uses a program compiler and programmer-designed test data to machine test the code for syntax errors (those detected at compile time) and logic errors (those detected at run time).
- This is usually the most rewarding step in the program development process.
- If the program has been well designed, the time-wasting frustration and despair often associated with program testing are reduced to a minimum.
- This step may need to be performed several times until you are satisfied that the program is running as required.

4.1. Steps in Program Development

7. Document and maintain the program

- Program documentation should not be listed as the last step in the program development process, as it is really an ongoing task from the initial definition of the problem to the final test result.
- Documentation includes both external documentation (such as hierarchy charts, the solution algorithm and test data results) and internal documentation that may have been coded in the program.
- Program maintenance refers to changes that may need to be made to a program throughout its life.
- Often, these changes are performed by a different programmer from the one who initially wrote the program.
- If the program has been well designed using structured programming techniques, the code will be seen as self-documenting, resulting in easier maintenance.