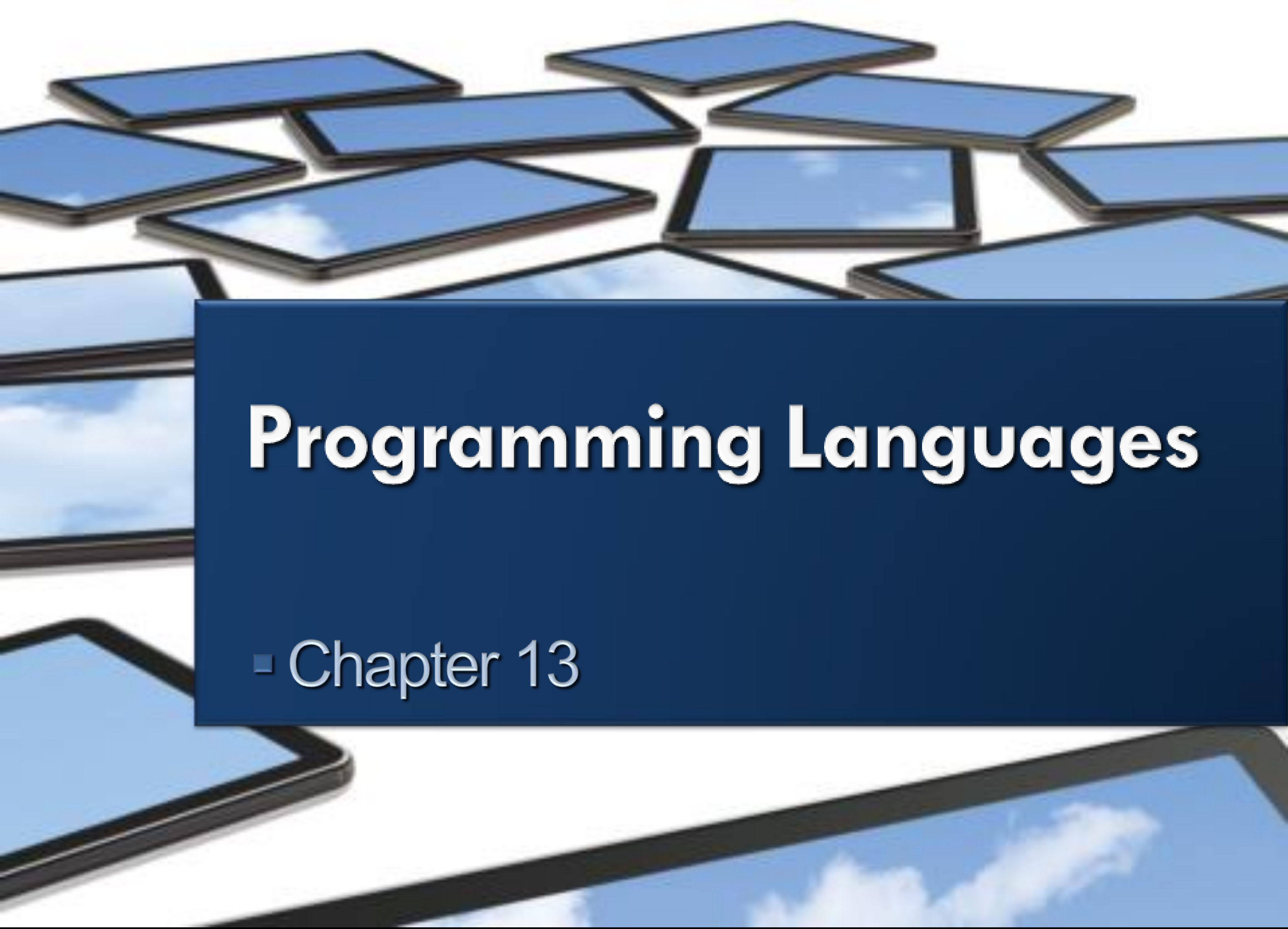


Programming Languages

- Chapter 13

Competencies (Page 1 of 2)

- Define programming of and describe the six steps of programming.
- Discuss design tools including top-down design, pseudocode, flowcharts, and logic structures.
- Describe program testing and the tools for finding and removing errors.

Competencies (Page 2 of 2)

- Describe CASE tools and object-oriented software development.
- Explain the five generations of programming languages.

Introduction

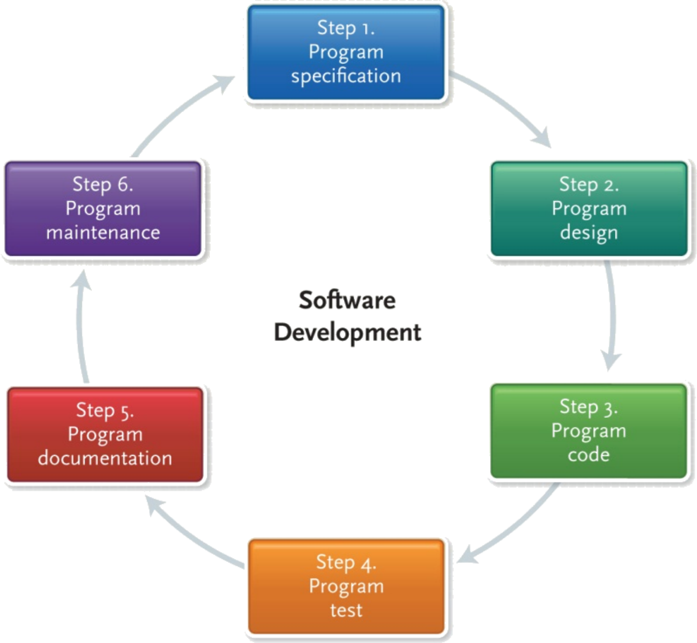
- In this chapter, you focus on Phase 4, Systems Development, of the systems life cycle and learn about the programming process and some of the programming languages that are available
- Competent end users need to understand the relationship between systems development and programming

Programs and Programming

- What is a program?
- A problem-solving procedure
  - A list of instructions
    - Prewritten
    - Custom-written
    - Application software
    - System software

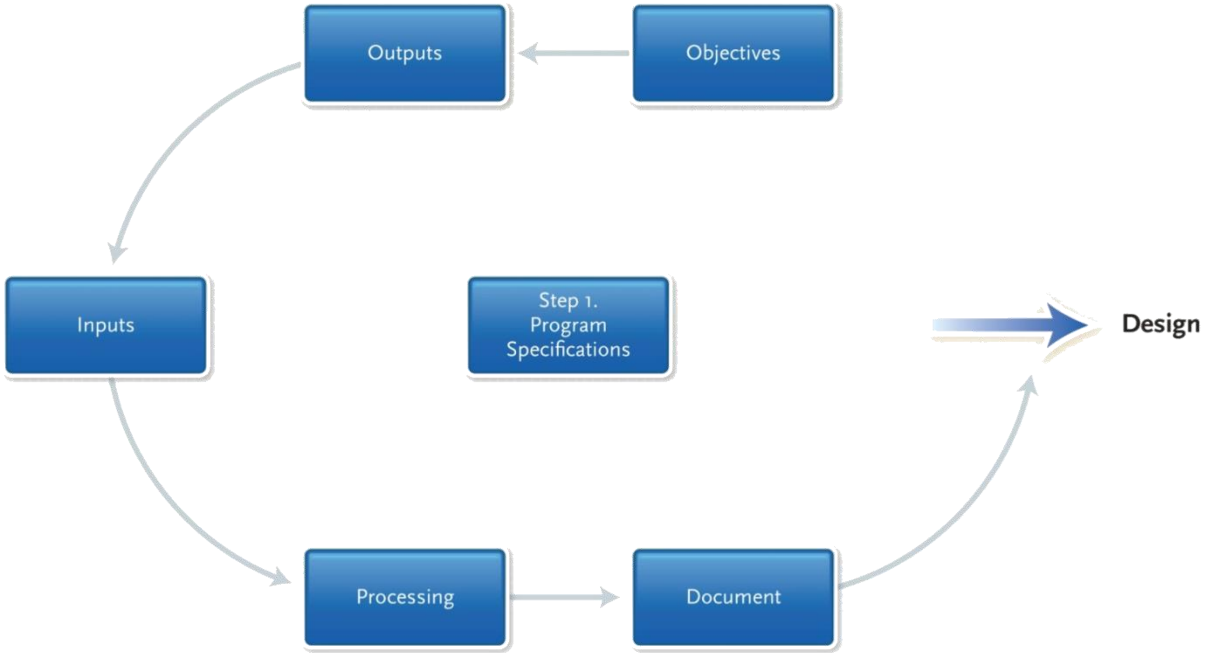
What is Programming?

- A list of instructions for the computer to follow to process data
- Software development
- Six step procedure
  1. Program specification
  2. Program design
  3. Program code
  4. Program test
  5. Program documentation
  6. Program maintenance



Step 1: Program Specification

- Also called program definition or program analysis
- Five steps to complete in the process



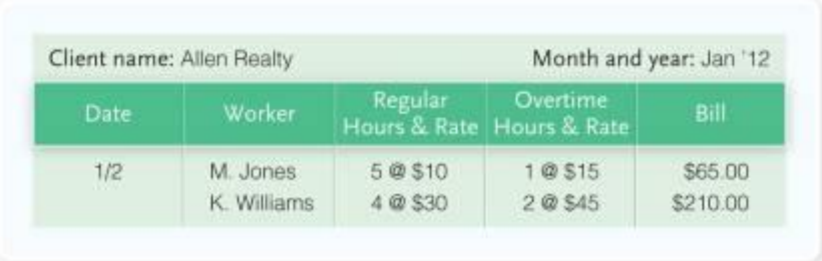
Program Objectives in Step 1

- Objectives are the problems that you are trying to solve
- Programming requires a clear statement of the problem that you are looking to address

Program Specification in Step 1

- Desired output
- Needed input data
- Required processing
- Documentation of program specifications

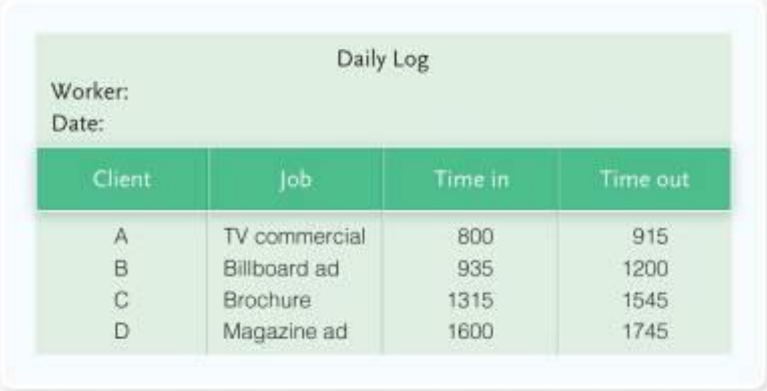
End user's sketch of desired output



Client name: Allen Realty   Month and year: Jan '12

Date	Worker	Regular Hours & Rate	Overtime Hours & Rate	Bill
1/2	M. Jones K. Williams	5 @ $10 4 @ $30	1 @ $15 2 @ $45	$65.00 $210.00

Example of input data



Daily Log

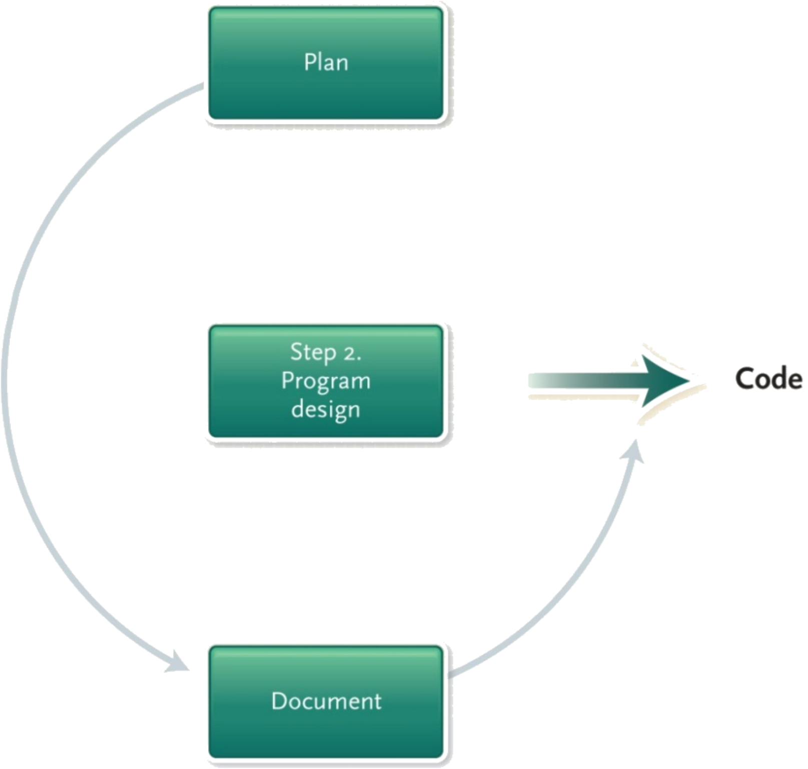
Worker:

Date:

Client	Job	Time in	Time out
A	TV commercial	800	915
B	Billboard ad	935	1200
C	Brochure	1315	1545
D	Magazine ad	1600	1745

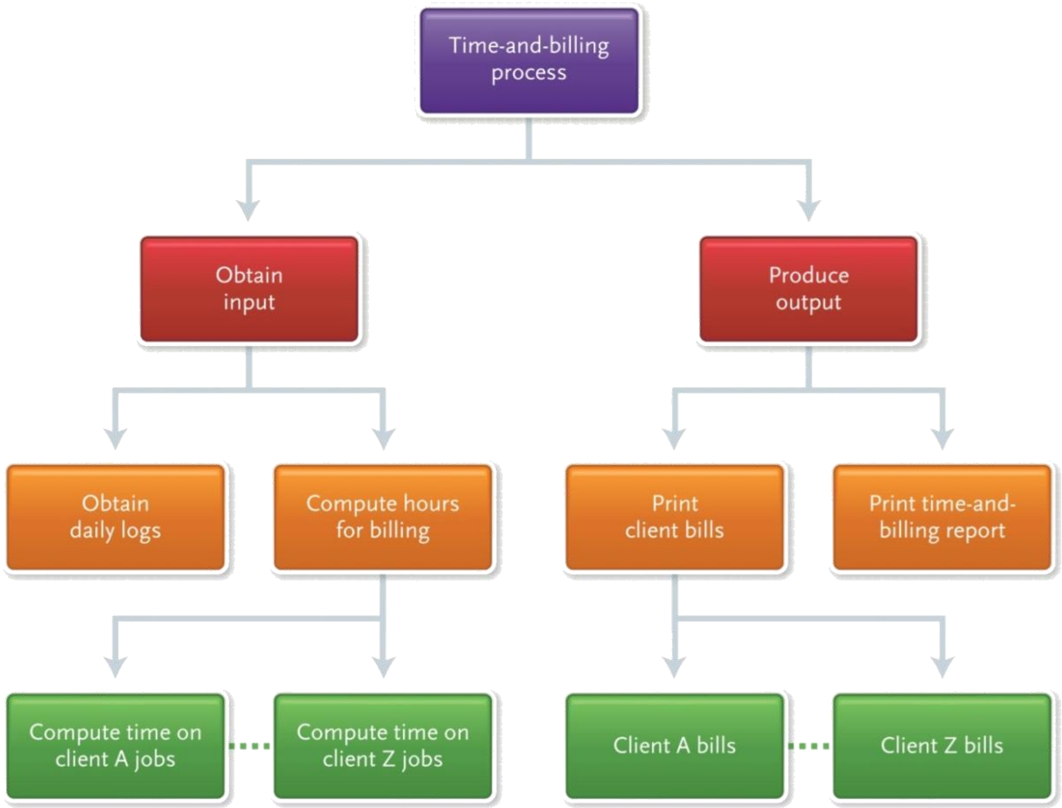
Step 2: Program Design

- Plan a solution using structured programming techniques
  - Top-down design
  - Pseudocode
  - Flowcharts
  - Logic structures



Top-Down Program Design

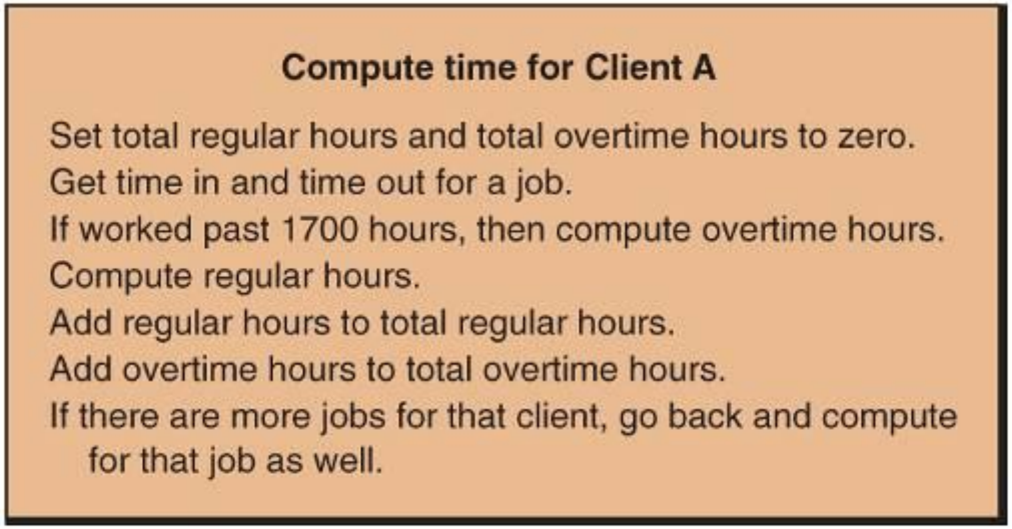
- Identify the program modules required



Return

Pseudocode

- Provide an outline of the logic and summarize the program you will write



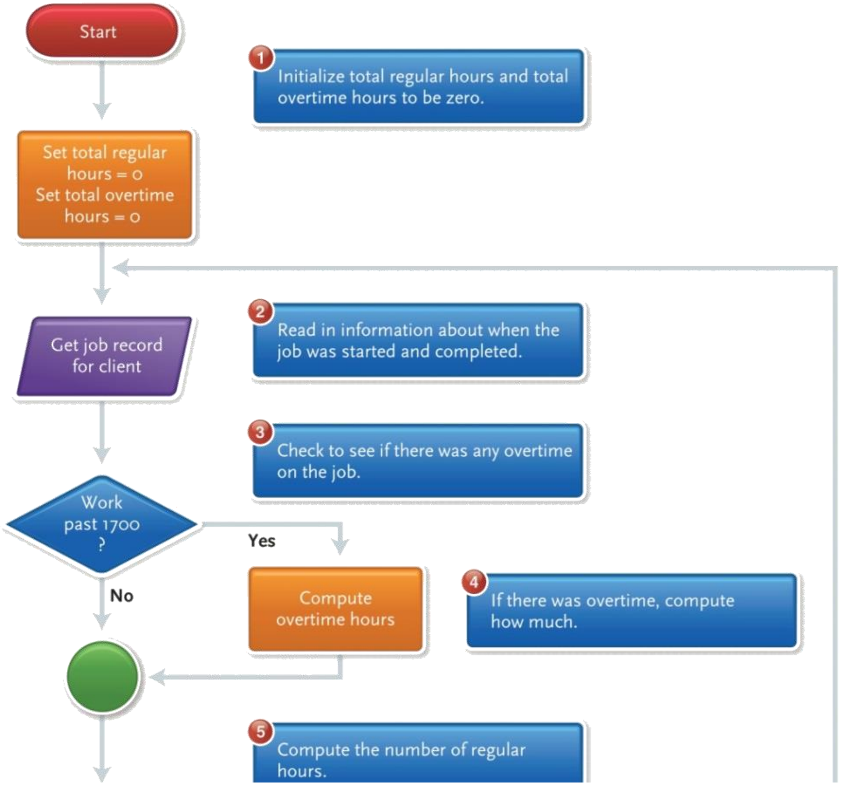
Compute time for Client A

```
Set total regular hours and total overtime hours to zero.
Get time in and time out for a job.
If worked past 1700 hours, then compute overtime hours.
Compute regular hours.
Add regular hours to total regular hours.
Add overtime hours to total overtime hours.
If there are more jobs for that client, go back and compute
    for that job as well.
```

Return

Flowcharts

- Graphically depict the sequence of steps required to solve a programming problem



Start

1 Initialize total regular hours and total overtime hours to be zero.

Set total regular hours = 0 Set total overtime hours = 0

2 Read in information about when the job was started and completed.

Get job record for client

3 Check to see if there was any overtime on the job.

Work past 1700 ?

Yes

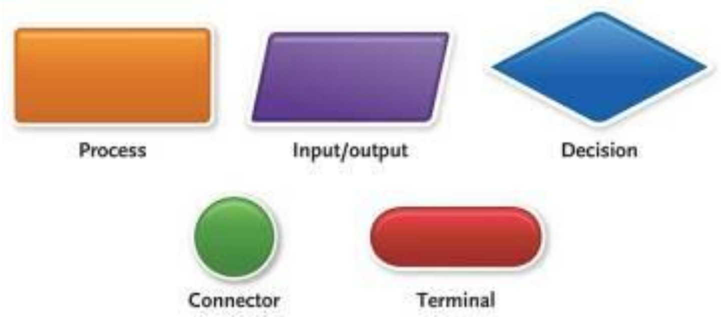
No

Compute overtime hours

4 If there was overtime, compute how much.

5 Compute the number of regular hours.

Flowchart Symbols

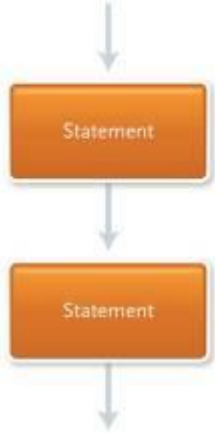


Return

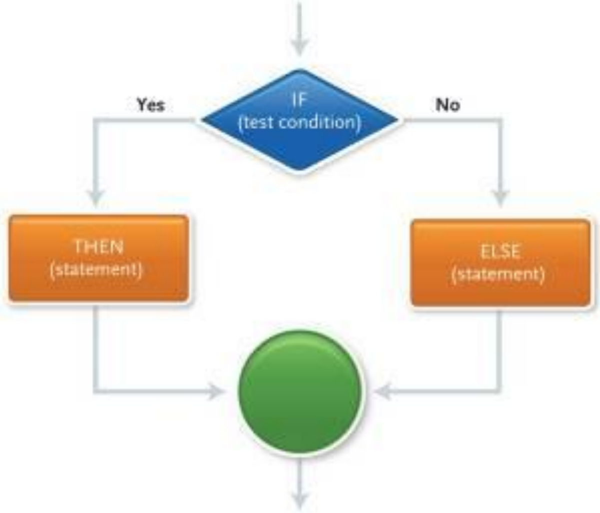
Logic Structures

- Concatenation structure
- Selection structure
- Repetition structure

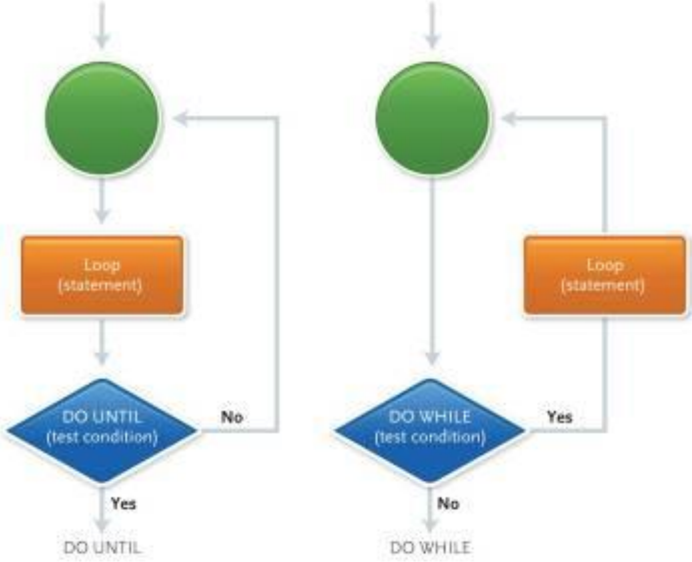
Concatenation



Selection



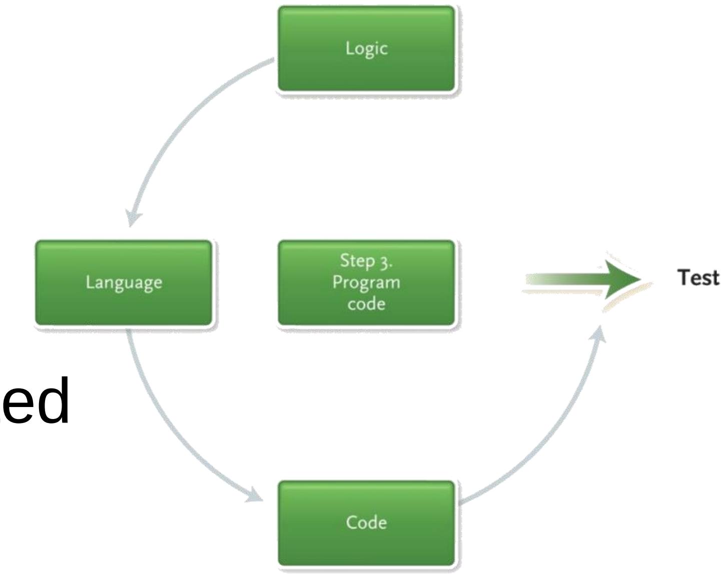
Repetition



Return

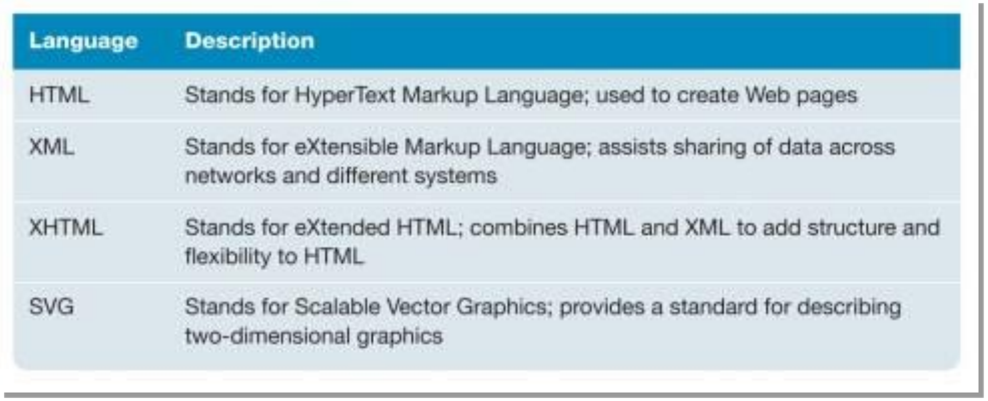
Step 3: Program Code

- Writing the program or coding
- Characteristics of a good program
  - Works reliably
  - Produces the correct output
  - Catches common input errors
  - Code is well-documented and understandable
  - Uses an appropriate computer language



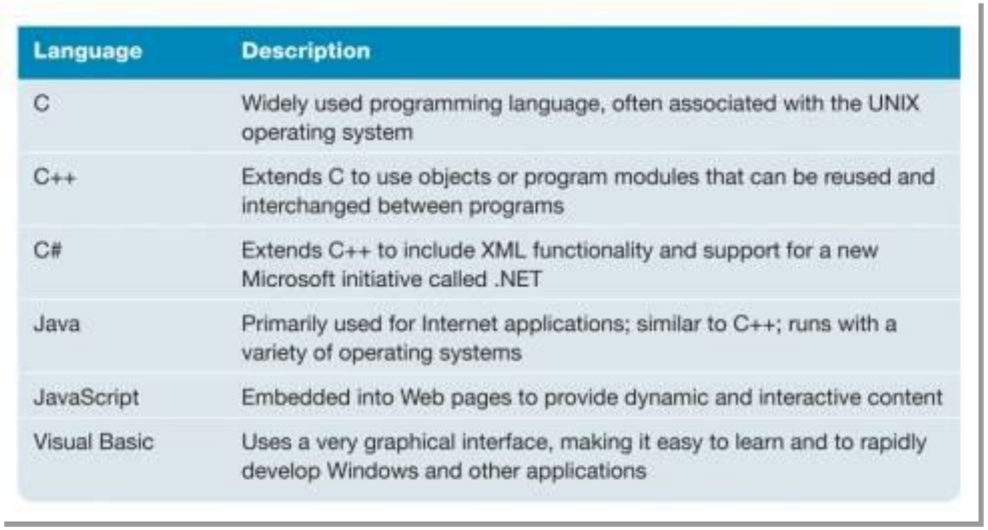
Coding in Step 3

- Content-markup languages



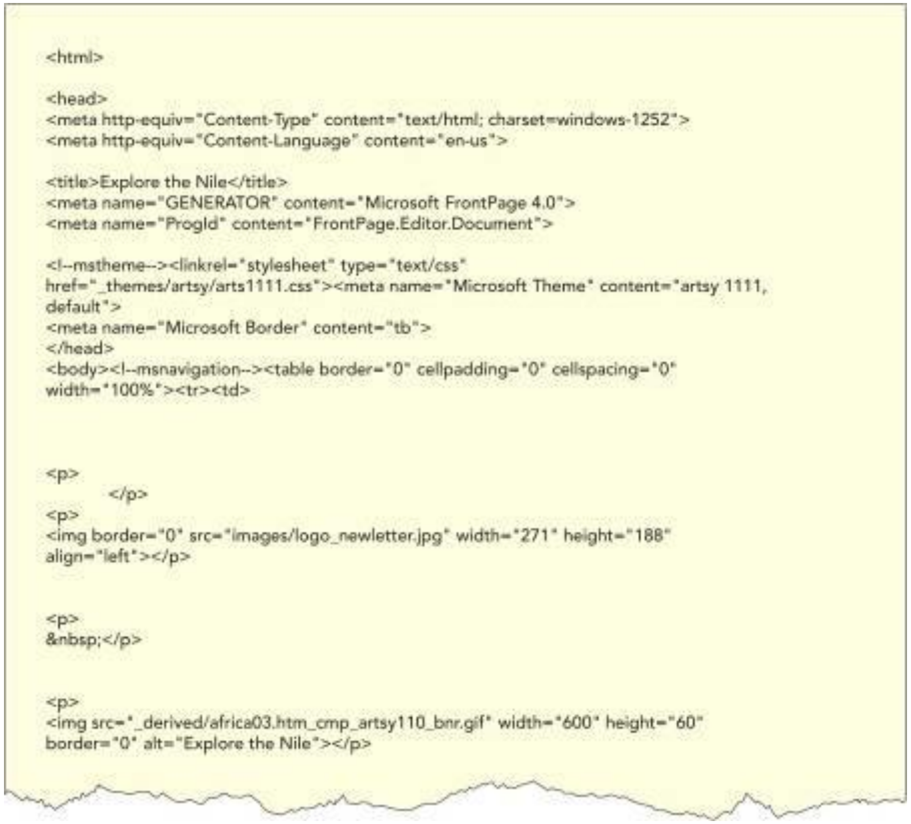
Language	Description
HTML	Stands for HyperText Markup Language; used to create Web pages
XML	Stands for eXtensible Markup Language; assists sharing of data across networks and different systems
XHTML	Stands for eXtended HTML; combines HTML and XML to add structure and flexibility to HTML
SVG	Stands for Scalable Vector Graphics; provides a standard for describing two-dimensional graphics

- Programming languages

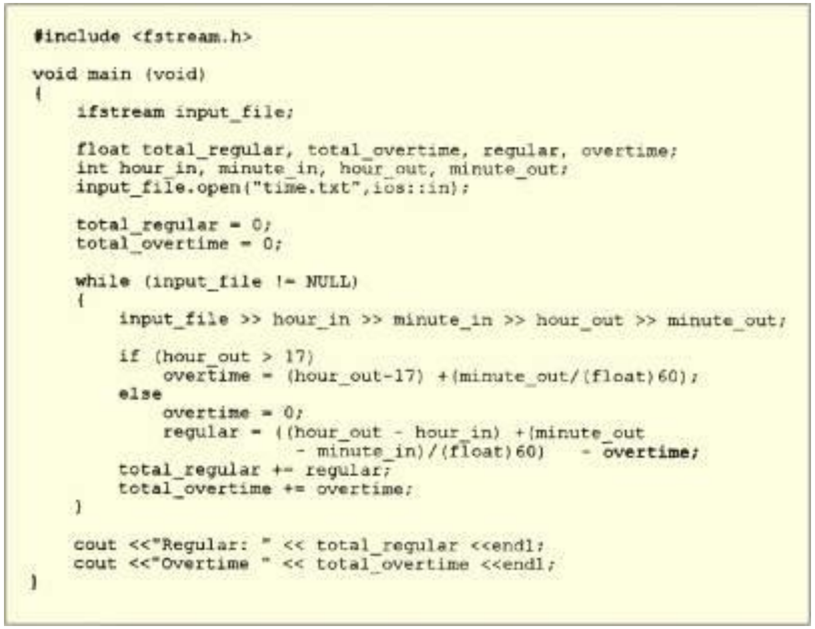


Language	Description
C	Widely used programming language, often associated with the UNIX operating system
C++	Extends C to use objects or program modules that can be reused and interchanged between programs
C#	Extends C++ to include XML functionality and support for a new Microsoft initiative called .NET
Java	Primarily used for Internet applications; similar to C++; runs with a variety of operating systems
JavaScript	Embedded into Web pages to provide dynamic and interactive content
Visual Basic	Uses a very graphical interface, making it easy to learn and to rapidly develop Windows and other applications

HTML and C++ Code



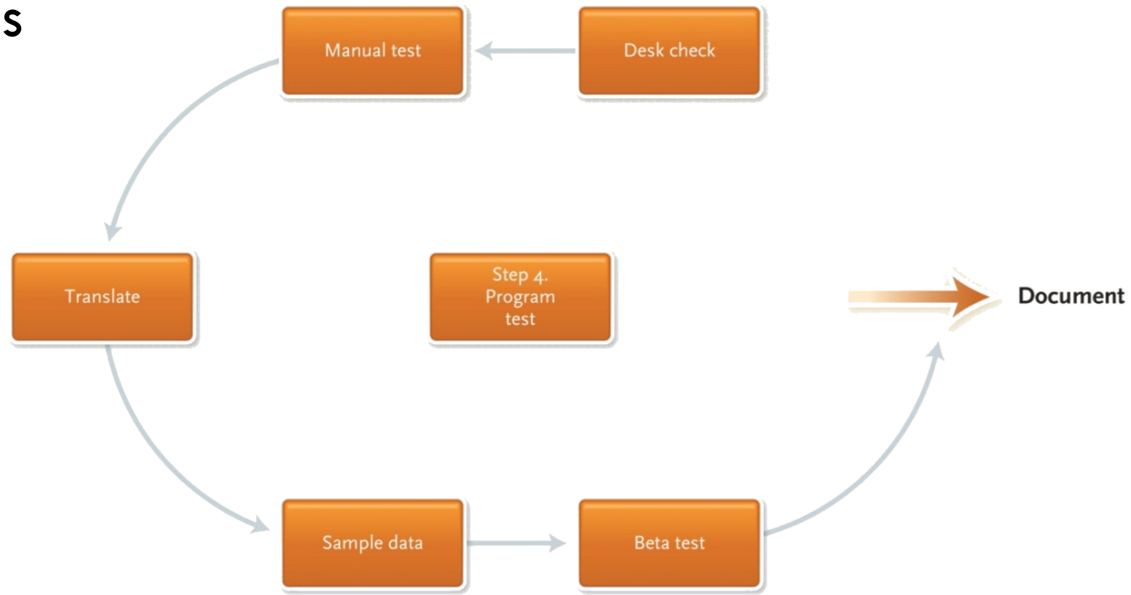
```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta http-equiv="Content-Language" content="en-us">

<title>Explore the Nile</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">

<!--mstheme--><linkrel="stylesheet" type="text/css"
href="_themes/artsy/arts1111.css"><meta name="Microsoft Theme" content="artsy 1111,
default">
<meta name="Microsoft Border" content="tb">
</head>
<body><!--msnavigation--><table border="0" cellpadding="0" cellspacing="0"
width="100%"><tr><td>



<p>
        </p>
<p>
<img border="0" src="images/logo_newletter.jpg" width="271" height="188"
align="left"></p>


<p>
&nbsp;</p>


<p>
<img src="_derived/africa03.htm_cmp_artsy110_bnr.gif" width="600" height="60"
border="0" alt="Explore the Nile"></p>
```



```
#include <fstream.h>

void main (void)
{
    ifstream input_file;

    float total_regular, total_overtime, regular, overtime;
    int hour_in, minute_in, hour_out, minute_out;
    input_file.open("time.txt",ios::in);

    total_regular = 0;
    total_overtime = 0;

    while (input_file != NULL)
    {
        input_file >> hour_in >> minute_in >> hour_out >> minute_out;

        if (hour_out > 17)
            overtime = (hour_out-17) +(minute_out/(float)60);
        else
            overtime = 0;
            regular = ((hour_out - hour_in) +(minute_out
                          - minute_in)/(float)60)    - overtime;
        total_regular += regular;
        total_overtime += overtime;
    }

    cout <<"Regular: " << total_regular <<endl;
    cout <<"Overtime " << total_overtime <<endl;
}
```

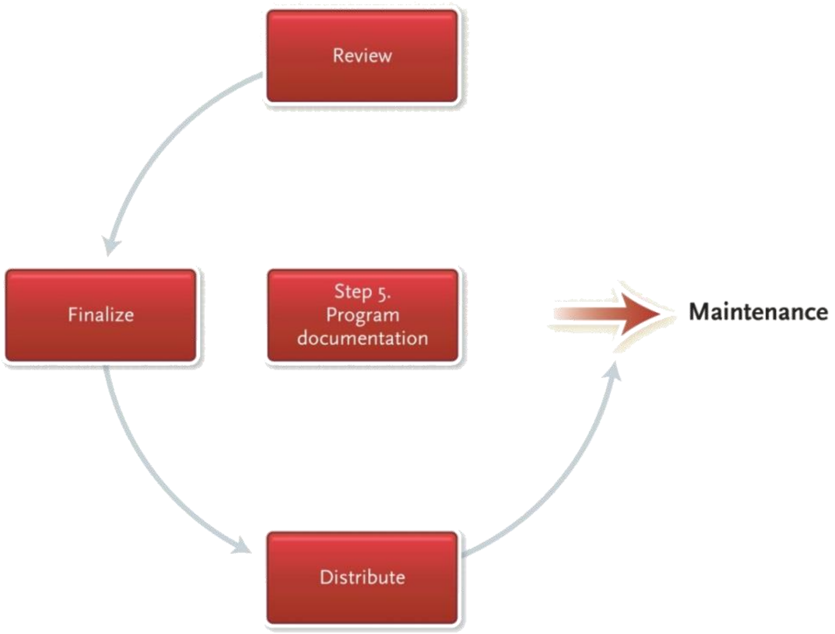
Step 4: Program Test

- Debugging to test code and eliminate errors
  - Syntax errors
  - Logic errors
  - Testing process



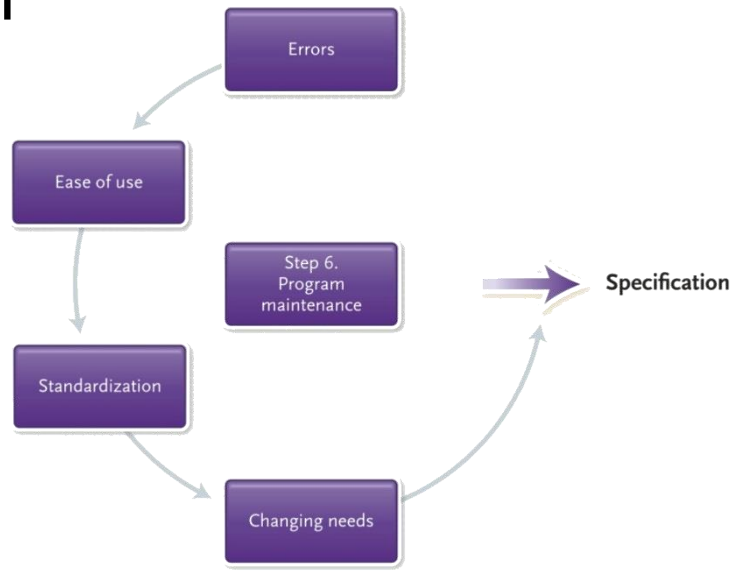
Step 5: Program Documentation

- Written descriptions about a program
- Important for people who will use and/or support the program
  - Users
  - Operators
  - Programmers



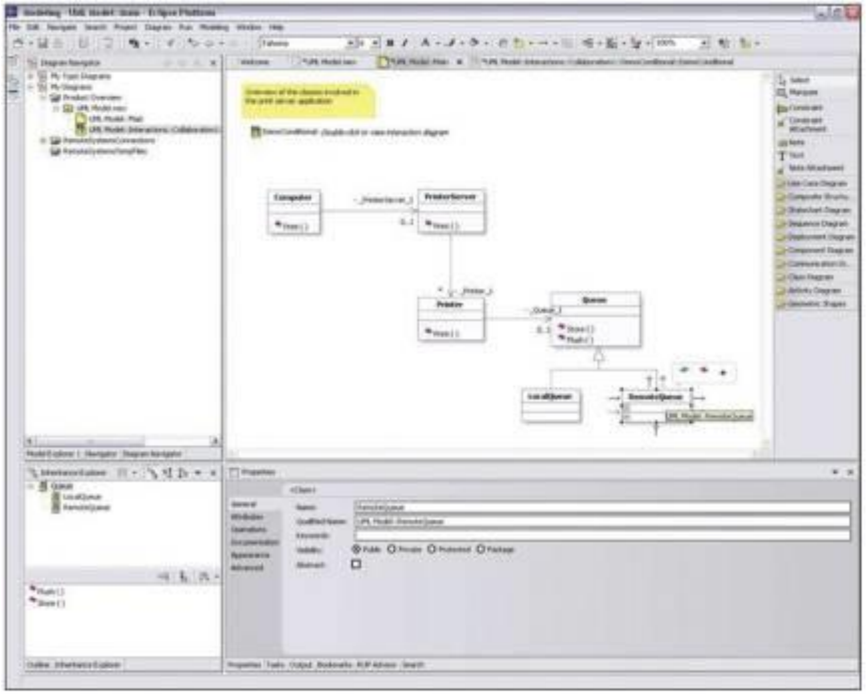
Step 6: Program Maintenance

- 75% of total lifetime cost
- Ensures program is
  - Error-free
  - Effective
  - Efficient
- Two activity categories
  - Operations
  - Changing needs



Case Tools

- Computer-aided software engineering (CASE)
  - Automates portions of the development process
    - Program design
    - Coding
    - Testing



OOP Software Development

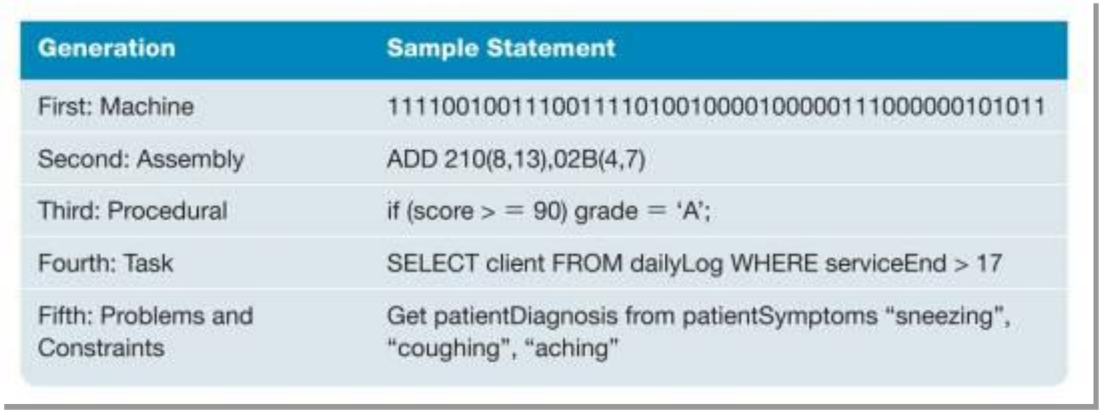
- Object-oriented programming (OOP)
- Focuses less on procedures, more on relationships between objects
- Objects contain both the data and the processing operations needed to perform a task

Generations of Programming Languages (Page 1 of 2)

- Occurring in “generations” or “levels”
  - Coding from machine languages to human or natural languages
- There are five distinct generations
  - Lower level is closer to machine language
  - Higher level is closer to human-like language

Generations of Programming Languages (Page 2 of 2)

- 1st Gen: Machine languages
- 2nd Gen: Assembly languages
- 3rd Gen: High level procedural languages (3GLs)
- 4th Gen: Task-oriented languages (4GLs)
- 5th Gen: Problem and Constraint languages (5GL)



Generation	Sample Statement
First: Machine	1111001001110011110100100001000001110000000101011
Second: Assembly	ADD 210(8,13),02B(4,7)
Third: Procedural	if (score > = 90) grade = 'A';
Fourth: Task	SELECT client FROM dailyLog WHERE serviceEnd > 17
Fifth: Problems and Constraints	Get patientDiagnosis from patientSymptoms "sneezing", "coughing", "aching"

Careers In IT

- Computer programmers create, test, troubleshoot, update and repair programs
- Employers seek individuals with
  - Degree in Computer Science or Information Systems
  - Desired traits include patience, logical thinking, and attention to detail

- Computer Programmers can expect to earn $51,500 to $88,000 annually

A Look to the Future

- Using a Wish List to Create a Program
  - Synapse Solutions has created a system called MI-tech that understands word order and meaning in English
  - Computer translates "wish list" into machine language

Open-Ended Questions (Page 1 of 2)

- Identify and discuss each of the six steps of programming.
- Describe CASE tools and OOP. How does CASE assist programmers?
- What is meant by "generation" in reference to programming languages? What is the difference between low-level and high-level languages?

Open-Ended Questions (Page 2 of 2)

- What is the difference between a compiler and an interpreter?
- What are logic structures? Describe the differences between the three logic types.