Booting a Computer

- When a computer is booted, it reads instructions from ROM (BIOS) to
  - Check what is connected
  - Test major components
  - Load Operating System

Operating System

- Operating System (OS) is responsible for
  - Receive User input (Keyboard, mouse, ...)
  - Manage allocation of memory
  - Process User input and provide output (Monitor, printer, ...)
  - Error handling
  - Read from and write to Disk
  - Allow execution of applications (Programs)
Major OSs

- MS- Windows
- Mac OS X
- UNIX
  - IBM AIX
  - HP-UX
  - SUN Solaris
  - LINUX
    - UBUNTU
    - PARDUS

Application Software

- Developed for a variety of purposes (Finance, Defense, Medical, Education, Entertainment, Manufacturing, ...)
- Written in variety of languages (Basic, C, Java, ...)
- Converted into (Compiled) machine language
- Executed on a computer with an appropriate OS
- Can be interactive
- Can be graphical
- Display output and/or store results
### Software Language

<table>
<thead>
<tr>
<th>Memory Address</th>
<th>Machine Language</th>
<th>Assembly Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>CLA</td>
</tr>
<tr>
<td>00000001</td>
<td>00010101</td>
<td>ADD X</td>
</tr>
<tr>
<td>00000010</td>
<td>00010101</td>
<td>ADD Y</td>
</tr>
<tr>
<td>00000011</td>
<td>00110101</td>
<td>STA Z</td>
</tr>
<tr>
<td>00000100</td>
<td>01110111</td>
<td>HLT</td>
</tr>
</tbody>
</table>

\[ Z = X + Y \]

<table>
<thead>
<tr>
<th>Memory Address</th>
<th>Contents</th>
<th>Variable Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000000</td>
<td>00000111</td>
<td>X</td>
</tr>
<tr>
<td>10000001</td>
<td>00001000</td>
<td>Y</td>
</tr>
<tr>
<td>10000010</td>
<td>00000000</td>
<td>Z</td>
</tr>
</tbody>
</table>

### High Level Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Origin Of Name</th>
<th>Application Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN</td>
<td>FORmula TRANslator</td>
<td>Science</td>
</tr>
<tr>
<td>LISP</td>
<td>LIS: Processor</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>COBOL</td>
<td>COmmon Business Oriented Language</td>
<td>Business</td>
</tr>
<tr>
<td>Ada</td>
<td>Ada Augusta Byron</td>
<td>Real-Time Dist. Systems</td>
</tr>
<tr>
<td>Prolog</td>
<td>Programming logic</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>Smalltalk</td>
<td>Objects talk to one another</td>
<td>GUI/OOP</td>
</tr>
<tr>
<td>C</td>
<td>Predecessor Language was B</td>
<td>General Purpose</td>
</tr>
<tr>
<td>C++</td>
<td>Successor of C (++) is increment operator</td>
<td>General Purpose/OOP</td>
</tr>
</tbody>
</table>
Software Development Method

Define Problem

Analyze

Design

Develop

Test

Maintain

Example

- Define Problem: Convert Pounds to Kilograms
- Analyze:
  - Input Pounds
  - Output Kilograms
  - Conversion 1 Pound = 0.454 Kilograms
- Design:
  - Ask user to provide weight in pounds
  - Convert weight into kilograms
  - Display the results
Example

- Develop

```c
/* Converts Pounds into Kilograms */
#include <stdio.h>
define KGS_PER_LBS 0.454 /* Conversion Constant */

int main (void)
{
    double pounds, /* input – weight in pounds */
        kgs; /* output – weight in kilograms */

    /* Get weight in Pounds */
    printf("Enter the weight in Pounds: ");
    scanf("%lf", &pounds);

    /* Convert weight into kilograms */
    kgs = KGS_PER_LBS * pounds;

    /* Display weight in kilograms */
    printf("The weight in kilograms is %f", kgs);
    return(0);
}
```

- Test:
  - Run the program for different inputs and check the output
    - 0, blank, negative, small/large, decimal, ...

- Maintain:
  - Resolve problems
  - Modify for new requests
Developing Programs

Example

- Create file with code
  - lbs2kgs.c
- Compile and link
  - gcc lbs2kgs.c –o lbs2kgs.exe
- Run
  - lbs2kgs
  - Enter the weight in Pounds: 180
  - The weight in kilograms is 81.72