Review Problem 1

Arithmetic (+, -, *, /, etc). What are the others? But they fall under a few basic types. One is...

Programming Languages have many instructions...
Stacks & local variables
Procedure calls
Floating point
Not our goal to teach complete assembly/machine language programming
Help figure out what the processor needs to be able to do
Understand the basics of assembly language

Typically one-to-one mapping to machine language
Simple, regular instructions - building blocks of C, Java & other languages
Assembly language

Green Reference card
Readings: 2.1-2.7, 2.9-2.10, 2.14

Assembly Language
Aside: C/C++ Primer
ADDA A, #2

SUBA A, #5

ADDI A, #2

ADDC A, #2

More complex: \( A = B + C + D - E \)

Simple format: easy to implement in hardware

```assembly
SUB <dst>, <src1>, <src2>
ADD <dst>, <src1>, <src2>
ADD <dst>, <src1>, <src2>

2nd operand
1st operand
Destination
Operator name

The basic instructions have four components:
```

ARM Assembly Language
All other operations work on registers. Load/store operation moves information between registers and main memory.

operands = operations

For capacity, computer has large memory (multi-GB).

For speed, CPU has 32 general-purpose registers for storing most operands.
<table>
<thead>
<tr>
<th>Comment</th>
<th>Function</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-op on write</td>
<td></td>
<td>X37</td>
</tr>
<tr>
<td></td>
<td>Return address</td>
<td>X30</td>
</tr>
<tr>
<td></td>
<td>Frame Pointer</td>
<td>X29</td>
</tr>
<tr>
<td></td>
<td>Stack Pointer</td>
<td>X28</td>
</tr>
<tr>
<td>Saved on call</td>
<td></td>
<td>X19-X27</td>
</tr>
<tr>
<td></td>
<td>Temporaries (saved across calls)</td>
<td>X16-X17</td>
</tr>
<tr>
<td></td>
<td>Don't use this</td>
<td>X18</td>
</tr>
<tr>
<td></td>
<td>Platform register</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linker scratch registers</td>
<td></td>
</tr>
<tr>
<td>Not saved on call</td>
<td></td>
<td>X9-X15</td>
</tr>
<tr>
<td></td>
<td>Volatile temporaries</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Result if a pointer</td>
<td>X8</td>
</tr>
<tr>
<td></td>
<td>Function arguments/ results</td>
<td>X0-X7</td>
</tr>
</tbody>
</table>

32 x 64-bit registers for operands