**Objects**

- These are basic computational entities in object oriented programming
  - Imagine you are building a house –
    - the bricks in the house are the basic entities,
    - no matter which room you are building, you need to use bricks
  - Imagine you are building a program
    - The basic entities are objects
    - Question: What are the basic entities in a procedural programming language like C?

**What are Objects?**

**Topics**

- Objects
- Classes
Exercise 1.1

- For each of the given problems, identify all the objects

Objects have Properties

- Objects are not just names
- They have properties.
  - Example:
    - Object: ObjectsFirstWithJavaBook
      - Property 1:
        - Property Name: Title
        - Property Type: String
        - Property Value: "Objects First with Java Book"
      - Property 2:
        - Property Name: DatePublished
        - Property Type: int (Short for Integer)
        - Property Value: 2005

More about Properties

- Other names for Properties:
  - Fields: This is commonly used in Java

- How to choose properties?
  - Depends on what you want to do with an object (Purpose)
    - Example:
      - ObjectsFirstWithJava object
        - Purpose: To print out details of the book
        - Good Properties: {Name, Publisher, Year}
        - Bad Properties: {J2SDK_version, authors_country, color_of_book}

Fields and Types

- Fields are places where the objects store data specific to the field.
  - Name: "John Galt"
  - SSN: 555-55-5555
  - Height: 6.1
- You can think of the fields as variables in other programming languages.
  - But these are local variables
    - Local to the objects
- NOTE: These fields already have values assigned to them
**Primitive Types in Java**

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Kind of Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>true or false</td>
</tr>
<tr>
<td>char</td>
<td>single character</td>
</tr>
<tr>
<td>byte</td>
<td>integer</td>
</tr>
<tr>
<td>short</td>
<td>integer</td>
</tr>
<tr>
<td>int</td>
<td>integer</td>
</tr>
<tr>
<td>long</td>
<td>integer</td>
</tr>
<tr>
<td>float</td>
<td>floating-point number</td>
</tr>
<tr>
<td>double</td>
<td>floating-point number</td>
</tr>
</tbody>
</table>

### Exercise 1.2

- For each of the given problems, identify the objects
- For each object, write the properties

### What are classes?

- Groups of objects that share common properties and common behaviors.
- How to assemble a class?
  - Collect all the objects from which you want to create a class
  - Identify common properties (fields)
  - Put these properties into class (without their values)
- If the objects do not share common properties, then it is not possible to form classes
- Now the class represents all the objects
Example of Class

- Example:
  - {Orange, Banana, Mango, Grape}
  - Fields: {Eatable, Taste, Nutritional_Value}
  - Class: Fruit

Exercise 1.3

- Given the statements form the class.

Methods in Classes

- Each method is like a function in procedural programming language like C.
  - The difference is that it is local to the class
  - It can act upon the local variables.

- Signature:
  - Access_Constraint Return_type Method_name arguments

- We will see the code in the lab