The Object Data Model
What is different:

-- object identity

-- complex structure

-- classes: attributes, methods, encapsulation

-- inheritance

-- more complex query languages
Object identity

-- built-in notion, similar to entities in ER model
-- independent of value (unlike in relational dbs)

Example

<table>
<thead>
<tr>
<th>parent</th>
<th>age</th>
<th>child</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>40</td>
<td>John</td>
<td>15</td>
</tr>
<tr>
<td>Sue</td>
<td>41</td>
<td>John</td>
<td>15</td>
</tr>
</tbody>
</table>

Same child?

With object identity:

Peter  → John
Sue    → John
Object identity

-- built-in notion, similar to entities in ER model
-- independent of value (unlike in relational dbs)

Example

<table>
<thead>
<tr>
<th>parent</th>
<th>age</th>
<th>child</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>40</td>
<td>John</td>
<td>15</td>
</tr>
<tr>
<td>Sue</td>
<td>41</td>
<td>John</td>
<td>15</td>
</tr>
</tbody>
</table>

Same child?

With object identity:

Peter → John

Sue → John
Class: type + extent

--extent: set of objects in class (like database instance)

--type: structure + methods

encapsulation: separate interface from implementation

Pseudo-code example:

class employee {
  string name;
  string address;
  int salary;
  int annual salary();
  string get-name();
  int employment-length();
};
Inheritance: class is a special case of given class

--what is inherited: structure + methods

Example

```java
class person {
    string name;
    string address;
}

class cutomer isa person {
    int credit-rating;
}

class employee isa person {
    date start-date;
    int salary;
}

class officer isa employee {
    int office-number;
    int expense-account-number;
}
```

Multiple inheritance: when one class is subclass of several classes -- potential ambiguity
Relationships

- Beside attributes, an object may have some relationships with other objects.
- These can be 1-1, 1-n or m-n relationships.
- Let us take an example:
Consider these classes:

Person  Course  Address

Student  Employee

Professor
Attributes and Methods

Person

Student

e takes, activities()

Employee

Professor

Course

Address

name, birth_date, age()

number, activities()

g grade

subject, notes()

street, city
Relationships

- Professor teaches Course
- Course taught_by Professor
- Student takes Course
- Course takes Student
- Person has_spouse Person
- Person has_address Address
- Person is_child_of Person
- Person is_parent_of Person
- Person has_prerequisite Course
- Course has_prerequisite Person
Graphic representation of more complex structure:
travel agency example