2.3 Records

Note_language_neutral

Grouping data

In the physical world, we are surrounded by basic items made from wood, metal, plastic, etc. But to keep the world underst think at a higher level, in terms of objects like an oven, car, or house. Similarly, keeping track of thousands of separate data thousands of variables in a program, is hard. A higher level approach is needed to organize data in a more understandable

Figure 2.3.1: Grouping data into higher level objects improves comprehension.

Source: Lumber yard (Olivier Colas / CC-BY-SA-3.0 via Wikimedia Commons), House (Ellin Beltz / Public domain via Wikimedia Commons)

A ***record*** (also called a ***struct***) declares a new type, which can be used to group multiple subitems. Ex: A patient's health da subitems such as height, weight, and age. Each record subitem is called a ***field***. Some programming languages refer to sub members. A subitem may be any type like int, char, or string.

PARTICIPATION

ACTIVITY 2.3.1: A record groups multiple data items.

Start 2x speed

healthData

height	62
weight	120
age	92

What is the patient's weight?

healthData.weight is 120

PARTICIPATION ACTIVITY 2.3.2: Naturally grouped data.

1) Select the pair indicating a flight's travel time.

- Hours and minutes
- Hours and cost
- Pounds and ounces

2) Select the group of items most likely to indicate the change provided to a person who pays for a meal.

- Ounce, gill, pint, quart, and gallon
- Mile, furlong, yard, feet, and inches
-

Dollars, quarters, dimes, nickels, and pennies

Accessing fields

`record healthData { ... }` defines a new data type named healthData. A programmer creates a record of type healthD variable declaration, as in the statement `healthData myVar;`. Fields can be accessed using ".", known as a ***dot notation*** op

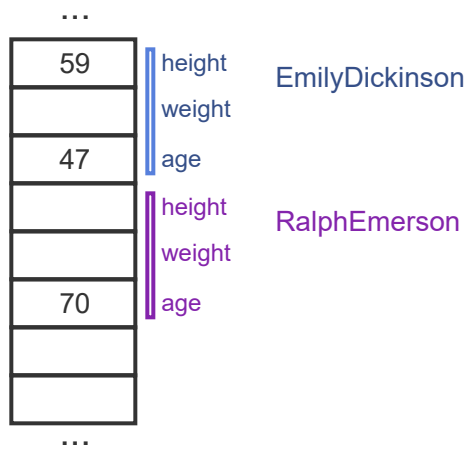
PARTICIPATION ACTIVITY 2.3.3: Using records in a program.

Start 2x speed

```
record healthData {
    int height;
    int weight;
    int age;
};


healthData EmilyDickinson;
healthData RalphEmerson;

EmilyDickinson.height = 59;
EmilyDickinson.age = 47;
RalphEmerson.age = 70;
```



PARTICIPATION ACTIVITY 2.3.4: Records and fields.

```
record houseStats {
   int bedrooms;
   float bathrooms;
   int garageSize;
};

houseStats houseA;
```

1) `houseStats` is a _____ .

- record
- field

2) The following statement assigns houseA's bedroom field to 5.

```
houseA.bedrooms = 5;
```

- True
- False

3) Fields within a record should be the same data type.

- True
- False

(*Note_language_neutral) This section is mostly language neutral

Provide feedback on this section