ITEC1620
Object-Based Programming

Lecture 11

User Defined Datatypes

Real-World Programming

- Determine the problem
- Define the problem
- Analyze the problem
- Solve the problem

A Case Study

- The Client
  - A credit card company
- The Assignment
  - Develop a support program to handle all of their credit card transactions

Determine the Problem

- What is happening?
  - Credit cards are issued
  - Purchases are made
  - Bills are sent out
- Note: Does not answer “What is the problem?” – clients usually know they have problems, not what they are

Define the Problem

- How much will be done?
  - Prototype?
  - Comprehensive system?
  - Comprehensive sub-component?
  - Random walk?

Analyze the Problem

- What is needed to issue a credit card?
  - Customer info, credit card info
- What is needed to handle a purchase?
  - Credit card info, purchase info
- What is needed to send out statements?
  - Customer info, credit card info

Solve the Problem

- Write (sub) programs to solve (sub) problems
- The easy part
  - This can be sent to China, India, Africa, …

Problem Modularization

- Break large problem into smaller problems
- Develop smaller programs for smaller problems
- Makes development (in teams) easier
- Makes debugging and maintenance easier – working on a smaller program

Problem Modularization II

- A computer is a machine that processes data
- Modularize problems by processes or data?
  - Processes → structured programming
  - Data → object-oriented programming

Structured Programming

- Decompose each task into its algorithmic components
  - Sequences
  - Branches
  - Loops

Processes Acting on Data

- Issuing cards
  - Write a process that initializes card data
- Handling purchases
  - Write a process that updates card data
- Sending statements
  - Write a process that accesses card data

Problems with Structured Programming

- Processes from all over affect card data
  - How to organize processes?
  - How to ensure that card data is secure?
- What is most important to a bank?
  - Record keeping → data
- Object-oriented design is preferred for most modern applications

Object-Oriented Analysis and Design

- Problem definition / OO analysis
  - What is a credit card?
  - What are the transactions?
  - How is a transaction handled?
  - Who makes the transactions?

Object-Oriented Analysis and Design II

- What are the model components?
  - Credit cards
  - Card holders
  - Transactions

Object-Oriented Analysis and Design III

- What are the component functions/responsibilities?
  - Transaction
    - ➢Update credit card balance
    - ➢Update credit card statement
    - ➢Update company record

Object Design

- Bottom-up
  - Identify all data elements and group them into logical sets
- Data
  - Employee ID, account number, company record, balance, credit limit, etc
- Credit card
  - Account number, balance, credit limit

Object Design II

- Top-down
  - Identify key model components and fill in their data elements
- Model Components
  - Credit cards, card holders, transactions
- Card holder
  - Name, address, credit rating, etc

Objects

- CreditCard
  - Fields (data)
    - accountHolderInformation
    - balance
    - creditLimit
    - transactionsList

Objects II

- AccountHolderInformation
  - Fields (data)
    - name
    - address
    - phoneNumber
    - personalID (e.g. SIN)
    - accountNumber

Objects III

- Transaction
  - Fields (data)
    - ➢date
    - ➢amount
    - ➢description

Objects IV

- CreditCard
  - Methods (actions)
    - ➢getBalance()
    - ➢getAvailableCredit()
    - ➢makePurchase()
    - ➢makePayment()
    - ➢// get/update information

Primitive Data and Objects

- Primitive data are primitive!
- Want to group pieces of data into larger, coherent sets that better represent the real world
- Groups of data are objects
  - Templates for data are classes
  - Known as “structures” in structured programming languages

Example

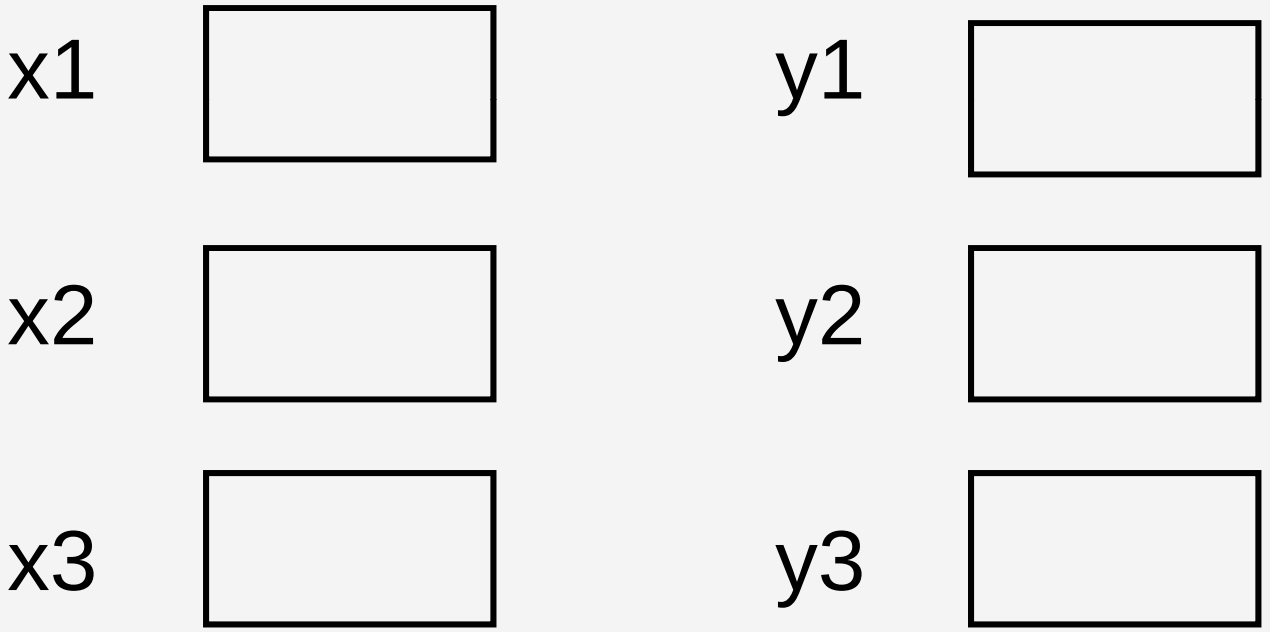
- Input the coordinates for a triangle
  - x1, x2, x3
  - y1, y2, y3
- The groupings are (x1, y1), (x2, y2), (x3, y3) – not (x1, x2, x3), (y1, y2, y3)
- How can we make this clear?

Example II

- Create a “Point” object
- Each Point has an x and a y coordinate
  - point1.x, point1.y
  - point2.x, point2.y
  - point3.x, point3.y

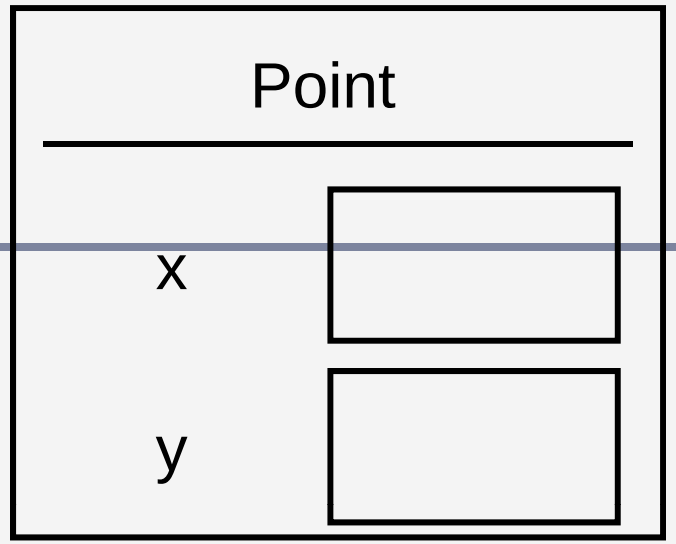
Example III

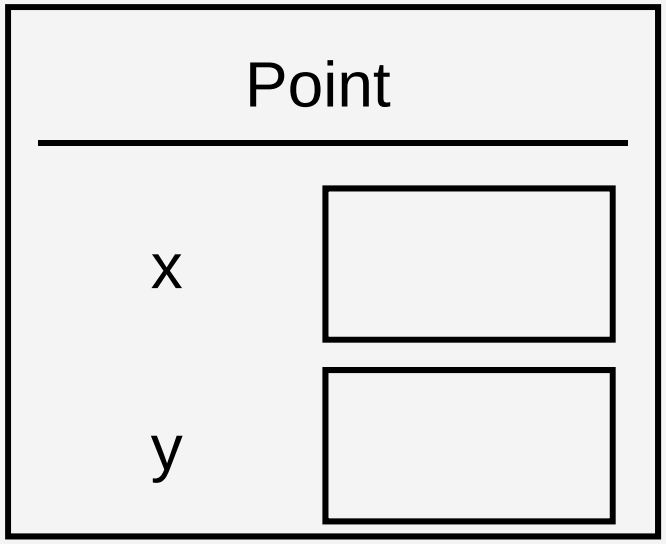
- Primitive data

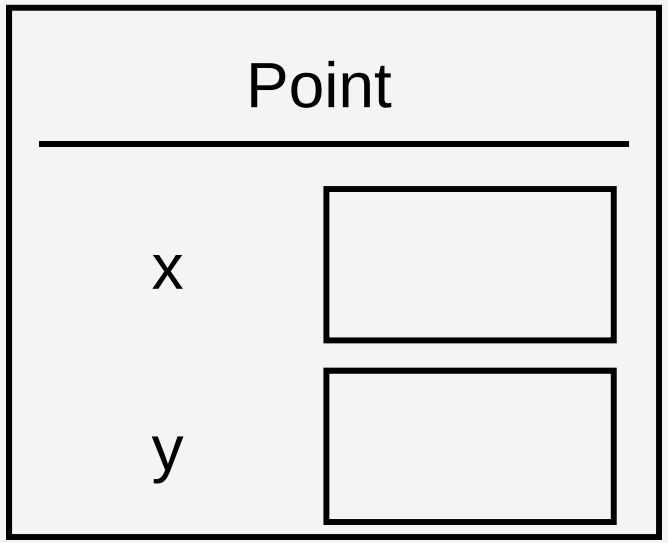


Example IV

- User-Defined Data







Classes and Objects

- Classes are a template for objects
  - Build classes in ITEC2610
  - Use classes in ITEC1620
- Concepts before code

Readings and Assignments