Study Objectives

• Finish Last Class Materials for Alternate Development Methodologies
• Alternate Development Methods
Alternate Software Development Methods

- What are alternative software development methods

  A. These are software programming or software engineering methods other than Structured methods
  B. These are independent of Software Development methodologies
  C. Object-Oriented Programming
  D. Data-oriented systems development
  E. Component-based model
  F. All of the above
Alternate Software Development Methods

- What are alternative software development methods

A. These are software programming or software engineering methods other than Structured methods
B. These are independent of Software Development methodologies
C. Object-Oriented Programming
D. Data-oriented systems development
E. Component-based model
F. All of the above
Structure Method of System Development

- Traditional Method
- Programs written in terms for function or subroutine
- A function can call another function
- Predates Object-Oriented (OO) System Development, but still in use
- “C” and “Pascal” are examples of programming languages which are not OO
Structure Method – Example

Making Pizza

1. Making Pizza

1.1 Get Grocery
1.2 Make Dough
1.3 Serve
1.4 Bake

1.1.1 Drive to Acme
1.1.2 Pick up Dough, Cheese, Sauce
1.1.3 Pay
1.4 Drive back Home

Context Diagram

Level 1 Data Flow Diagram

Get Grocery
Make Dough
Serve
Bake

Drive to Acme
Pick up Dough, Cheese, Sauce
Pay

Drive back Home
Structure Programming – Pseudocode Example

MakePizza (5, [mushroom, chicken, pepper])

GetGrocery (int howmany, list toppings) {
  Read the list of toppings;
  Read howmany pizza;
  Drive to supermarket;
  Pick-up grocery;
  Pay;
  Drive back home;
  return;
}

Writing Structured Programming

MakePizza (int count, list toppings) {
  GetGrocery (count, toppings);
  MakeDough ();
  Bake ();
  Serve ();
}

More Structured Analysis Details can be found at:
https://en.wikipedia.org/wiki/Structured_analysis
Alternate Development Method – Object-Oriented System Development

- Object-Oriented Programming Languages (OOP Language)
  - Simula67 (1967); Smalltalk (1970s)
  - C++, Object Pascal, Ada95 (1980s – 1990s)
  - Java (1990s); C# (2000s)

- Objects and Class
  - Objects are constructs with “attributes” and the “methods” that acted on it
  - Before OOP, “Structured” programming was used, where a program was written as a “function” that had a number of instructions that did the things like adding numbers, sort words, retrieve message from email etc.
  - In OOP the same problem is looked at differently. For example, Numbers, Words, or emails can be thought as objects. The objects have attributes like value of the numbers, the words, or the emails. There are “methods” associated with the objects, which act upon the attributes of the objects.
  - Objects can interact with each-other by sending messages.
  - Similar objects with different values can be created by the template of the object, call “class,” which is similar to cookie-cutter
Object-Oriented System Development

• Advantages of OOP
  – Data Encapsulation:
    • allows private attributes of the object not visible outside the object (abstraction)
    • allows implementation of the method to be separated from the signature of the method
  – Inheritance: allows reuse by creating super class and subclass (hierarchy)
Example of OOP using Java

public class Human
{
    private String name = "no name"; // the name of this human
    private Human friend = null; // the human's friend

    //This "creates" a new Human

    public Human(String name, Human friend) {
        this.name = name;
        this.friend = friend;
    }

    public Human(String name) {
        this.name = name;
        this.friend = null;
    }

    public Human() {
        this.name = "no name";
        this.friend = null;
    }

    public void sayName() {
        System.out.println("My name is " + this.name);
    }

    public void sayGoodnight() {
        if (friend == null)
            System.out.println("Good night nobody.");
        else
            System.out.println("Good night " + friend.name);
    }
}

public class Main
{
    public static void main(String[] args) {
        //create a new human object john
        Human john = new Human("John");
        //create a human object named jane with john as a friend
        Human jane = new Human("Jane", John);

        john.sayName(); //shows 'My name is John'
        john.sayGoodnight(); //shows 'Good night nobody.'

        jane.sayName(); // shows 'My name is Jane'
        jane.sayGoodnight(); //shows 'Good night John'
    }
}

Source: Adapted from http://simple.wikipedia.org/wiki/Object-oriented_programming
Example of Class and Objects

Class Human

Human
- name
- Human friend;
- sayName()
- sayGoodnight()

Human Jane
- name "Jane"
- friend "John"
- sayName()
- sayGoodnight()

Human John
- name "John"
- friend ""
- sayName()
- sayGoodnight()

Object Jane
- name = "Jane"
- friend = "John"
- sayName()
- sayGoodnight()

Object John
- name = "John"
- sayName()
- sayGoodbuy()
Inheritance

Example of Inheritance

Class Human

Man: Human
- String name
- Human friend
- String beard
- sayName()
- sayGoodnight()
- sayBeardcolor()

Woman: Human
- String name
- Human friend
- Boolean married
- Human husband
- sayName()
- sayGoodnight()
- Boolean IsMarried()
OO Concept

• A good reference (page 1-32)

Alternative Development Methods – Component Based Development

• Component based development is an outgrowth of object-oriented development

• Component types
  1. Stand-alone client component: applications that expose services to other software such as MS Word, Excel
  2. In-process client component: run from a container such as web browser such as java applet
  3. Stand-alone server components: run on the server, and can be invoked using some kind of remote procedure calls such as CORBA, Sun’s RMI, Microsoft DCOM
  4. In-process server components: run on the server within containers such as Sun’s EJB or Microsoft MTS
Component Based Development

• Advantages
  – Reduces development time
  – Promotes modularity
  – Promotes reuse
  – Reduces development cost
  – Supports multiple development environments
Modern Component Based Distributed Architecture (week 7)

Client-Server Computing Model had typically multiple clients connecting to the same server that processes the request.
Alternative Development Methods – Web-Based Development

• Web-services based development further simplifies the integration across the components using platform and language independent standards

• Web-services characteristics
  1. XML based
  2. SOAP (Simple Object Access Protocol) defines a standard way of invoking remote object
  3. WSDL (Web Services Description Language) is XML-based representation of the input and output data, along with the service contract to be invoked
  4. UDDI (Universal Description, Discovery, and Integration) is used for the service discovery

• Web-services based integration is today’s de-facto standard
Alternative Development Methods – Software Reengineering

• Software Reengineering allows revamping an existing system by extracting and reusing business logic and program component

• Similar to Business Process Re-engineering (BPR)
Alternative Development Methods – Reverse Engineering

• Involves extracting the business logic and design from the code
• Decompilers are tools used in reverse engineering
• Helps in faster delivery
• Risks involves any current license agreement prohibiting reverse engineering
Question

Which of the following development methodologies is based on knowledge in someone's head, as opposed to traditional requirements?

A. System Development Life Cycle (SDLC)
B. Object-Oriented Programming (OOP)
C. Agile
D. Rapid Application Development (RAD)
Question

Which of the following development methodologies is based on knowledge in someone's head, as opposed to traditional requirements?

A. System Development Life Cycle (SDLC)
B. Object-Oriented Programming (OOP)
C. Agile
D. Rapid Application Development (RAD)
Upcoming Assignments/Tests

Group Project -3 (Testing): Thu 4/14 before the class

Questions?
Summary of Today’s Class

• OOP
• Alternate Development Methods
• Focus of the Next Class and Reading
• Questions