**TTYP – C++ revisited 1**

Which of the following statements are reasonable after the following statement: `char* fred = new char[5];`

- a. `fred = bill;`
- b. `strcpy(fred, "store");`
- c. `strcpy("store",fred);`
- d. `fred = "store";`

**TTYP – C++ revisited 2**

Which statements are reasonable immediately after the following statement: `char* max;`

- a. `max = "fred";`
- b. `max = new char[10];`
- c. `max = 0;`

---

**Scenarios, Use Cases, Interactions**

- The lab this week will cover creating use case diagrams and interaction diagrams in Rational Rose.

**Use Case Diagram**

- Draw important actors
- Add each action

**Draw Interaction Diagrams**

- Vertical lines for the classes
- Use scenario to put in function calls
**TTYP – interaction diagram**

- Draw a interaction diagram for checking out a book

---

**Software Development**

Software Crisis, Software Quality and the Software Life Cycle

Pressman pp 1-52

---

**Software Crisis**

- Software is delivered over-budget, late, and not correct
  - software complexity
  - programmer productivity
- DeMarco’s (IEEE Software Apr 94)
  - only too costly because they would like to pay less
  - only late because they want it sooner

---

**Future of Software**

- increasing complexity
- increasing criticality
- trends
  - more formalisms
  - object-oriented
  - Reuse, COTS, and components
  - software measurement
  - emphasis on quality

---

**Quality - the ultimate goal**

Software that does what it is suppose to or
I will know it when I see it

---

**How to achieve quality**

- Top-down
  - through software life cycle
  - through quality improvement
  - see Dilbert
- Bottom-up
  - Personal Software Process
Software Life Cycle (SLC)

formalizing the process of software development
Pr:22-53

Waterfall Model

Waterfall

Royce - 1970
first realization that there was a standard sequence of tasks
useful as a management tool
  – milestones
maybe too restrictive

Software vs Houses

Requirements
  – Requirements
  – Design
  – Implementation
  – Maintenance

Schematic
  – Blueprints
  – Building House
  – Home Repairs

Design

Requirements
  – Phases
    – Preliminary Analysis
    – Market Analysis
    – Requirements Specification

Documents
  – Statement of Work (SOW)
  – Requirement Specification (SRS)
    » Data Flow Diagram
    » Data Specification
  – User Manual, Test Plans, SQA Plans

Phases
  – Architectural Design
  – Detailed Design

Documents
  – Hierarchy Diagram
  – Module Specification
  – PDL
Implementation

- Phases
  - coding
  - unit testing
  - integration
  - acceptance
  - delivery

- Documents
  - code
  - test reports
  - final user manual

Maintenance

- Tasks
  - error correction
  - enhancement
  - version control
  - regression testing
  - version release

- Documents
  - error report
  - test reports
  - revised documentation

Waterfall Alternatives

- Rapid Prototyping
- Spiral Model
- Incremental Development
- RUDE

Rapid Prototyping

- Build a throw-away version
  - test concepts
  - test requirements
  - not complete functionality
- Saves developing wrong product
- Saves developing unnecessary features

Spiral Model

- Boehm (IEEE Computer May88)
- Revisit the basic tasks of
  - determine objectives
  - evaluate alternatives
  - develop
  - plan next phase
- uses prototypes

Incremental

- Parnas (IEEE TOSE Mar79)
- Design minimal subset
- Design minimal increments
- Always able to deliver something useful
RUDE - the AI approach

- Run
- Understand
- Debug
- Enhance