

# Exercise

```
( 1)      DOUBLE PRECISION FUNCTION SIN(X, E)
( 2) C      THIS DECLARATION COMPUTES SIN(X) TO ACCURACY E
( 3)      DOUBLE PRECISION E, TERM, SUM
( 4)      REAL X
( 5)      TERM=X
( 6)      DO 20 I=3,100,2
( 7)      TERM=TERM*X**2/(I*(I-1))
( 8)      IF (TERM.LT.E) GO TO 30
( 9)      SUM=SUM+(-1**(I/2))*TERM
(10)      20 CONTINUE
(11)      30 SIN=SUM
(12)      RETURN
(13)      END
```

- .LT. is less than
- \*\* is exponentiation
- C is comment to end of line
- CONTINUE is an empty statement
- DO is the for loop  
lno var=begin, end,  
incr
- Return value is held in function name
- Default INTEGER type for variables beginning with i-n; otherwise REAL

# Defects

- (1) Can't get double precision result from single precision argument
- (2) "DECLARATION" -> "FUNCTION"
- (2) "SIN (X) TO" -> "SIN (X) TO"
- (5.5) SUM is not initialized
- (2.5) Should mention Maclaurin series
- (2.5) Should actually list series and give reference
- $\sin x = 1 - x^{**2}/3! + x^{**4}/5! + \dots$
- (9) "(-1\*\*..." -> "((-1)\*\*..."
- (9) Exponentiation of -1 should just alternate signs
- (4.5) Doesn't work on negative values
- (7) Strength reduction on "X\*\*2"
- (7) Constant folding on "X\*\*2" (XSQ = X\*X)
- (7) Strength reduction on "I\*(I-1)"
- (8-9) Test and exit should come after accumulation

# Inspections

- Also called *reviews* and *walkthroughs*
- Complements other verification techniques such as testing and proofs
- Cost-effective way of finding faults
- Redundant inspections by different teams may find even more faults

# Purpose

- Detection of *faults (bugs or defects)* not *errors or failures*
- Adherence to standards
- Not education of staff
- Not problem solving or fault correction
- Not status reporting

# Reviewable Artifacts

- Requirements documents
- Specifications
- Architectural designs
- Detail designs
- New code
- Fixes
- Test plans
- Documentation

# Inspection Process

- **Planning:** select participants, schedule, assign roles, partition activity, organize and distribute materials (-5 days)
- **Preparation:** individual study noting potential defects (~10 pages of text or ~100 LOC per hour)
- **Inspection:** (same rate; two hours maximum)
- **Rework:** confirm and remove defects
- **Follow-up:** verify corrections; analyze results; update database; improve process

# Roles

- **Moderator:** (see next slide)
- **Recorder:** records detected issues on form during reading
- **Reader:** paraphrases document systematically
  - Descriptive (no purpose attributed to code);
  - Third-party ("it" rather than "I")
- **Inspectors:** (3-6)

# Moderator Responsibilities

- Evaluates preparedness of inspection team prior to commencement
  - Aborts inspection if team is not prepared
- Evaluates entry criteria (actual completion of work to be inspected)
- Should be trained
- Must run meeting effectively and arbitrate differences
- Must be technically competent
  - Might not be expert on work being inspected
- Selects participants
- Responsible for completion of follow-up

# Recording

- Issue/question oriented (something to be looked into) rather than necessarily assuming that a problem exists
- Location; area
- Description
- Type (from a controlled vocabulary)
- Severity

# Inspection Meeting

- Introduction of participants
- Statement of objectives
- Evaluation of preparedness
- **Systematic inspection**
  - Paraphrase, scenarios, questions (verification conditions), or checklist
- Record results on inspection form
- Review and summarize detected faults
- Determine severity and priority
- Determine disposition responsibility

# Thoroughness

- The purpose of inspections is to find faults
- One key determinant of success is to have a deterministic measure of *thoroughness*
- Various approaches have been used
  - Systemmatic reading (probably most common)
    - Line coverage
  - Scenarios (use cases)
  - Checklist of error types
    - May be externally provided (common industrial experience)
    - Or based on historical data from that company
  - Verification conditions (Cleanroom)

# Checklist-Based Reviews

- More effective if each checklist item is verified separately
- Create checklist from previously detected defects
  - But if you are only looking for checklist items, you may miss items not on the list and never add them. So use defects detected during testing as well
- Typical list size is 15-20 items
- Retire items if they are not detected for six months

# Defect Classification

- Minor rework
  - Verified by author
- Conditional rework
  - Verified by moderator
- Major rework
  - Re-inspection is required if greater than 20% of document or 20 hours of work or 100 LOC have been affected

# Alternative Review Styles

- Fagan: formal (as described in these slides)
  - Staff time; scheduling overhead; training
- Pair programming
  - Complementary to formal reviews
- Tool-assisted reviews
  - Differencing, source control, distribution, display, annotation, review metrics, enforcement, integration with IDE, customization
- E-mail, pass-around reviews
  - Used for open-source projects
  - Pre or post source code management system control check in
- Over-the-shoulder reviews
  - Complete informality

# Guidelines

- Inspections should not be used for personnel evaluations
- Managers should not attend unless they have participated technically in the production of the document being inspected
- Don't treat review as an opportunity for training staff
  - Instead, hold a separate meeting
- Spread out inspections over time

# Guidelines - 2

- Do not allow additional participants (observers)
- Avoid use of “you” and discussion that might raise defensiveness
- Author should not be moderator, recorder, or reader (?)
- Each type of inspection should have its own checklist

# Guidelines - 3

- Avoid problem solving during the inspection meeting
- Cost of inspection is between 10-20% of development effort
- Avoid discussions of style
- Inspection as a *go/no-go* activity
- Inspectors' signatures on inspection report

# Guidelines - 4

- Limits: smaller of 250 lines or two hours
  - Defect detection drops off significantly thereafter
- There is some evidence to suggest that lightweight reviews are more effective in finding defects per staff-hour
  - But harder to enforce formality
  - Author annotation/desk checking; 2 or 3 people
- Defect detection rates vary dramatically depending on review goals and code complexity

# Guidelines - 5

- Ego effect: the knowledge that an author's code will be reviewed improves the quality of the delivered code
- Improve coding skills by looking at other people's code
- Dangers: damaged egos; big-brother effect
- Use checklists to detect omissions
- Typical yield for normal reviews is 70-90%

# Process Data

- Kind of review
- Artifact
- Date/time
- Participants
- Preparation time
- Meeting time

- Issues detected
- Types
- Severity
- Effectiveness questionnaire
- Stored in a database

# Metrics

- Inspection rate: kLOC/staff-hour
- Defect rate: defects/staff-hour
- Defect density: defects/kLOC
- Yield:  $(\text{detected} / \text{total}) * 100$