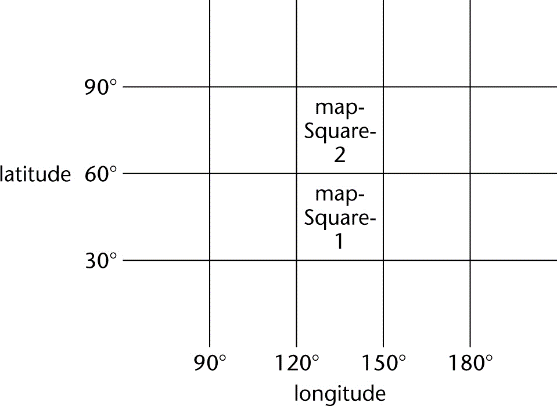
**1) Can you explain the advantages of C++, Java and Python respectively**

These three programming languages are the most popular among the coders in terms of competitive coding and programming. C++ as of today in its efficiency, speed, and memory make it widely popular among coders. Java is platform independent. It continues to add considerable value to the world of software development. Python requires less typing, provides new libraries, fast prototyping, and several other new features

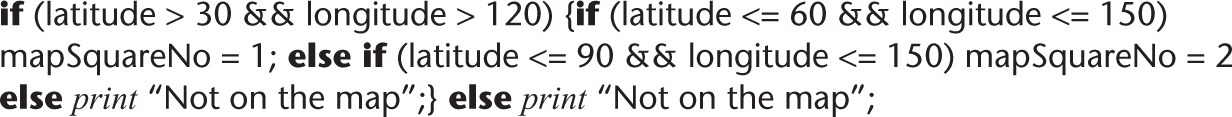
**2. Please write a code example to resolve the issue of hard-coded file name.**

**3. How to change if statements nested to a depth of greater than three ? Give an example**

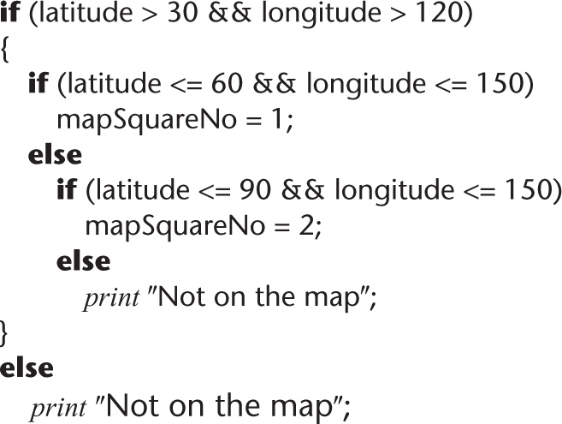
Example - A map consists of two squares. Write code to determine whether a point on the Earth’s surface lies in mapSquare1 or mapSquare2, or is not on the map



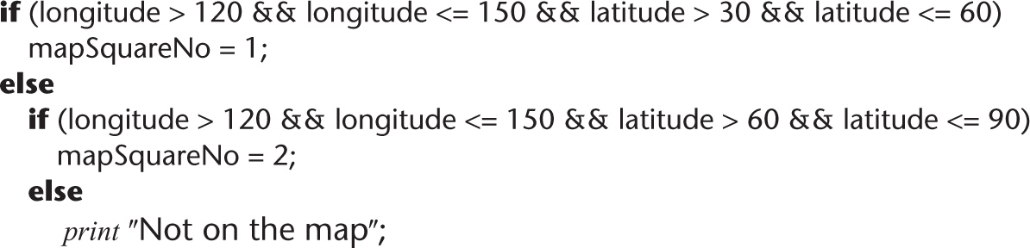
Solution 1. Badly formatted



Solution 2. Well-formatted, badly constructed



Solution 3. Acceptably nested



**4) Explain the difference between logic artifacts and operational artifacts**

Logic artifacts include the decision-making flow of control

In the example, artifacts a, b, c, d, g, j

Operational artifacts perform the actual operations of the product

In the example, artifacts e, f, h, i, k, l, m

The logic artifacts are developed before the operational artifacts

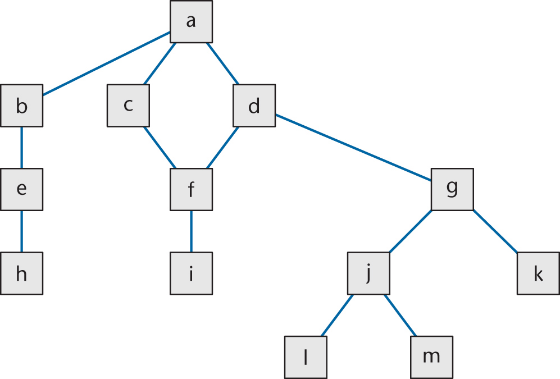
**5) What is fault isolation ?**

fault isolation (If the product as a whole is tested against a specific test case and the product fails, then Fault isolation - A previously successful test case fails when mNew is added to what has been tested so far. The fault must lie in mNew or the interface(s) between mNew and the rest of the product

**6) What is top down integration, bottom up integration and sandwich integration ?**

**Top down integration**

If code artifact mAbove sends a message to artifact mBelow, then mAbove is implemented and integrated before mBelow - One possible top-down ordering is a, b, c, d, e, f, g,h, i, j, k, l ,m

* + 
* Another possible top-down ordering is

a

[a] b, e, h

[a] c ,d, f, i

[a, d] g, j, k, l, m

Bottom up integration

If code artifact mAbove calls code artifact mBelow, then mBelow is implemented and integrated before mAbove One possible bottom-up ordering is l, m, h, i, j, k, e,f, g, b, c, d, a

* Another possible bottom-up ordering is

h, e, b

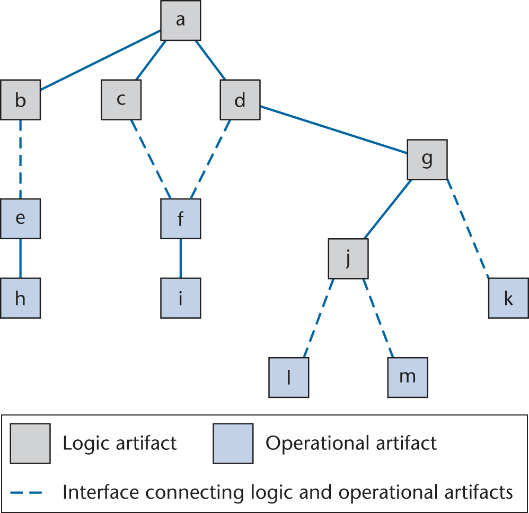
i, f, c, d

l, m, j, k, g [d]

a [b, c, d]

**Sandwich Integration**

* Logic artifacts are integrated top-down
* Operational artifacts are integrated bottom-up
* Finally, the interfaces between the two groups are tested



**7) What are the advantages and disadvantages of the three integration methods**

**Top down integration**

Advantage 1: Fault isolation A previously successful test case fails when mNew is added to what has been tested so far. The fault must lie in mNew or the interface(s) between mNew and the rest of the product

Advantage 2: Stubs are not wasted. Each stub is expanded into the corresponding complete artifact at the appropriate step

Advantage 3: Major design flaws show up early

* + Logic artifacts include the decision-making flow of control
    - In the example, artifacts a, b, c, d, g, j
  + Operational artifacts perform the actual operations of the product
    - In the example, artifacts e, f, h, i, k, l, m
  + The logic artifacts are developed before the operational artifacts

Problem 1 - Reusable artifacts are not properly tested and Lower level (operational) artifacts are not tested frequently

**Bottom-up Integration**

Advantage 1 - Operational artifacts are thoroughly tested

Advantage 2 - Operational artifacts are tested with drivers, not by fault shielding, defensively programmed artifacts

Advantage 3 - Fault isolation

Difficulty 1 - Major design faults are detected late

Solution - Combine top-down and bottom-up strategies making use of their strengths and minimizing their weaknesses

**Sandwich Integration**

Advantage 1 - Major design faults are caught early

Advantage 2 - Operational artifacts are thoroughly tested

* + They may be reused with confidence

Advantage 3 - There is fault isolation at all times

**8) What are two extremes to testing ?**

* + *Test to specifications* (also called black-box, data-driven, functional, or input/output driven testing) - Ignore the code — use the specifications to select test cases

*-Test to code* (also called glass-box, logic-driven, structured, or path-oriented testing) - Ignore the specifications — use the code to select test cases

**9) What is the art of testing**

Select a small, manageable set of test cases to

Maximize the chances of detecting a fault, while

Minimizing the chances of wasting a test case

**10) What are the equivalence classes defined by range (10, 100) ? Please select test cases for this**

**11) What are the three kinds of unit-testing techniques ? Please compare them**

Black-box testing, Glass-box testing, Reviews

* + [Myers, 1978] 59 highly experienced programmers - All three methods were equally effective in finding faults
  + [Hwang, 1981] All three methods were equally effective
  + [Basili and Selby, 1987] 42 advanced students in two groups, 32 professional programmers
  + Advanced students, group 1 - No significant difference between the three methods
  + Advanced students, group 2 - Code reading and black-box testing were equally good and Both outperformed glass-box testing

**12) What are product testing and acceptance testing ? Please compare them**

Product testing for COTS software Alpha, beta testing

Product testing for custom software

* + The SQA group must ensure that the product passes the acceptance test
  + Failing an acceptance test has bad consequences for the development organization

The client determines whether the product satisfies its specifications

Acceptance testing is performed by

* + The client organization, or
  + The SQA team in the presence of client representatives, or
  + An independent SQA team hired by the client

The key difference between product testing and acceptance testing is

* + Acceptance testing is performed on actual data
  + Product testing is preformed on test data, which can never be real, by definition

**13) What is postdelivery maintenance**

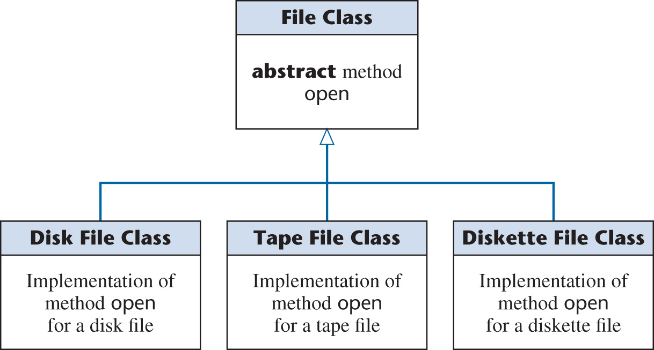
* + *Any* change to *any* component of the product (including documentation) after it has passed the acceptance test

**14) What are corrective, adaptive and perfective maintenance**

* Corrective maintenance
  + To correct residual faults
    - Analysis, design, implementation, documentation, or any other type of faults
* Perfective maintenance
  + Client requests changes to improve product effectiveness
    - Add additional functionality
    - Make product run faster
    - Improve maintainability
* Adaptive maintenance
  + Responses to changes in the environment in which the product operates
    - The product is ported to a new compiler, operating system, and/or hardware
    - A change to the tax code
    - 9-digit ZIP codes

**15) Why polymorphism and dynamic binding can have negative effects on maintenance ?**

The code is hard to understand if there are multiple possibilities for a specific method



* The product fails on the invocation myFile.open ()
* Which version of open contains the fault?
  + A CASE tool cannot help (static tool)
  + We must trace