Abstraction

Generalisation of ideas rather than manifestation

Control Flow

Sequencers influence flow of control

Jumps transfer control to anywhere in the program

Escapes sequencers that transfer control out of block commands

Exceptions signal and handle abrnormalities

Sequencers are loops and if statements

Jumps implemented by keyword goto x: C++ support

Escapes terminate execution keyword break in Java

Can halt program entirely C++ exit()

Exceptions reroute errors with prompts and turnabouts

Throw try catch blocks

Procedural Abstraction

Proper/Function

Function evaluates expression and returns result

Called by expressions

Proper procedure contains command to executed and updates variables. Observable but not for implementation

Abstraction Principle

Possibility of designing procedures that abstract over any syntactic category provided the constructs have some computation.

Variable Access - call procedure and result in some variable

Declarations - Generic Units are abstract over these - Templates in C++ and class templates In Java

Parameter Passing Mechanisms

Copy Parameter binds formal parameter to local variable

Reference Parameter binds formal parameter to the argument itself

Copy Parameter Mechanisms

Copy in local variable created and initiliased - value parameter

Copy out local variable created not initialised - result parameters

Copy in Copy out local varialbe created, initialised - value-result parameters

Parameter Passing Mechanisms

Constant Parameter - value bound to formal parameter

Variable parameter - variable bound to formal parameter

Procedure Parameter - procedure bound to formal parameter

Correspondence Principle

Constant corresponds to constant definition

Variable corresponds to renaming definition

Copy-In parameter corresponds to initialised variable declaration

Remove any unnecessary differences between declarations and parameter passing mechanisms - Correspondence Principle

For each form of declaration there exists a corresponding parameter passing mechanism

Haskell follows correspondence Principle if restricted to first order values

C++ supports copy-in parameters and variable parameters so limited extent