AS CS Revision Notes P2

Tony Wu 20296

Chapter 0 Methods of Answering

Notes: if the given points is in 1,2,3,etc then its a mark scheme answer. For mark scheme answer, you must answer the exact point to be rewarded the point.

Chapter 9 Algorithm Design and Problem Solving

Loops
FOR = counter controlled
WHILE = pre conditional
DO WHILE /REPEAT UNTIL = post conditional

Library Routine
Program libraries store pre-written functions and routines
The program library can be referenced/imported
The functions/routines can be called in her own program

Library Routines Benefits
1 They are tried and tested so free from errors
2 They perform a function that you may not be able to program yourself
(for example encryption)
3 They are readily available / speed up development time

Abstraction
Abstraction is used to filter out information / data that is not necessary for the task // To keep only information / data that is necessary for the task

Abstraction Benefits
The time of required to develop the program is reduced
The program is smaller in size so take up less pace in memory
Customer satisfaction is greater as their requirements are met without any extraneous features

Decomposition
Break a complex problem into smaller parts that can be further subdivided into even smaller parts until each part is easy to understand and solved.

Decomposition Benefits
1 Breaking a complex problem down makes it easier to understand / solve //
smaller problems are easier to understand / solve
2 Smaller problems are easier to program / test / maintain
3 Sub-problems can be given to different teams / programmers with different
expertise // can be solved separately

Identifier Table (2pts)
Name of identifier
A description of what the identifier is used for / the purpose of the identifier
The data type of the identifier
The number of elements of an array // the length of a string

An example data value
Value of any constants used
The scope of the variable (local or global)

Chapter 10 (Abstract) Data Structures

Stack
Each stack element contains one data item
A Pointer to the front / start of the stack
Data is added at front / start and removed from front / start // works on a LAFO basis
May be circular

Queue
1 Each queue element contains one data item
2 A Pointer to the front / start of the queue
3 A Pointer to the back / end of the queue
4 Data is added at back / end and removed from front / start // works on a FIFO basis
5 May be circular

Linked List
1 Each node contains data and a pointer to the next node
2 A Pointer to the start of the list
3 Last node in the list has a null pointer
4 Data may be added / removed by manipulating pointers (not moving data)
5 Nodes are traversed in a specific sequence
6 Unused nodes are stored on a free list // a free-list pointer to the Free List

Record Benefits
1 Array of records can store mixed data types / multiple data types under a single identifer
2 Tighter / closer association between ErrCode and ErrText // simpler code as fields may be referenced together // values cannot get out of step as with two arrays
3 Program easier to design / write / debug / test / maintain / understand

Array Benefits
1 Algorithm to process / search / organise the data is easier to implement // Values may be accessed via a loop-controlled variable / iterated through using index
2 Makes the program easier to design / code / test / understand
3 Multiple instances referenced via a single identifier / so fewer identifiers needed // Easier to amend the program when the number of students increases

Linked List > Array
1 Pointers determine the ordering of data // only the pointers need to be changed when data changed
2 Easier to add / delete data (to maintain correct sequence) in a linked list // description of moving data to maintain correct sequence when array used

Array > Linked List
1 Need to store pointers as well as data
2 More complex (to setup / implement)

Chapter 12 Software Development

Development Cycle
Analysis
Feasibility study
Investigation
Fact finding
Design
Structure chart
Pseudo code
Flowchart
Coding
Testing
Maintenance

Waterfall
Suitable for small project with short timescale
+ Easy to manage and use
+ Each stage is deliverable
+ Overlapping and predicable
- Difficult to change at later stage
- Working program is produced late in life cycle

RAD Rapid Application Development
Suitable for large projects
+ reduce overall development time
+ Rapid frequent customer feedback
+ Easy to do modifications
- stem under development needs to be modular
- Needs strong teams of skilled developers

Iterative
Suitable for large projects
+ easier to debug and test program
+ More flexible as easier to alter requirements
+ Working program developed quickly at early stages
- not suitable for simple project
- Needs good planning overall and for every stage

Program Testing
Syntax error, logic error, runtime error
Walkthrough/Dry run
White box testing
Black box testing
Stub testing (dummy function that does nothing)

AS CS Revision Notes P2

Tony Wu 20296

Maintenance
Corrective, perfective, adaptive

Alpha Testing
In house testing tested by programmers
Programmer will check that the software works as required
The development team will record problems that occur
The problems identified will be addressed

Beta testing
1 Testing carried out by a small group of (potential) users
2 Users will check that the software works as required / works in the real world / does not contain errors
3 Users will feedback problems / suggestions for improvement
Acceptance Testing
Testing carried out by a the public
The public will check that the software works as required
The public will record problems that occur and give suggestions doer improvement
The problems identified will be addressed

Whitebox testing
Detailed testing how each procedure work
Read the program to consider the structure and logic line by line

Blackbox testing
Testing a module's input and output
Do not examine the internal logic and structure of the program

Stub Testing
1 A simple module is written to replace each of the modules.
2 The simple module will return an expected value // will output a message to show they have been called

Integration Testing
1 Modules that have already been tested individually
2 are combined into a single program which is then tested as a whole

Testing Data
Example: 0<=x<=12
Normal: x = 10
Abnormal x = 15
Extreme x = 12
Boundary x = 12 and x = 13