1.2.3 Software development

OCR A-Level Computer Science (H446) - H446/01 - Exam questions and mark schemes

3 past paper questions and 5 practice questions, 38 marks in total. Answer each question in the space provided, then check your work against the mark schemes at the end. compsciwizard.com

Question 1 H446/01, June 2022 [9 marks]

A programmer has been asked by a client to create a complex computer program. Compare the spiral model and waterfall lifecycle methodologies for this task. You should include: how both methodologies could be used to develop a complex computer program, the benefits of each methodology for this task, the drawbacks of each methodology for this task.

Question 2 H446/01, June 2019 [4 marks]

A company makes anti-virus software. Anti-virus software is an example of a utility. In order to keep up to date with the latest virus threats, the company is continually updating their software. The programmers use an Extreme Programming approach when developing the updates. Explain what is meant by Extreme Programming and why it is a suitable approach in this case.

Question 3 H446/01, June 2017 [4 marks]

A software development team is writing a word game. The team is using Rapid Application Development. Describe the Rapid Application Development process.

Question 4 Practice question (H446/01 style) [4 marks]

A small company wants a stock-control system. Its requirements have been fully documented and are unlikely to change. Explain why the waterfall lifecycle is suitable for this project and describe one drawback of using it.

Question 5 Practice question (H446/01 style) [4 marks]

Describe two practices used in extreme programming (XP) and explain how each one improves the quality of the code produced.

Question 6 Practice question (H446/01 style) [6 marks]

A games studio is developing a new game and expects the requirements to change frequently as players test early versions.

Compare rapid application development (RAD) and the waterfall lifecycle for this project, and recommend which the studio should use.

Question 7 Practice question (H446/01 style) [3 marks]

Explain why the spiral model is suited to large projects where the risks are not fully understood at the start.

Question 8 Practice question (H446/01 style) [4 marks]

An algorithm is written in pseudocode:

```
01 total = 0
02 for i = 1 to 4
03   total = total + i * i
04 next i
05 print(total)
```

Complete a trace table showing the value of i and total after each iteration of the loop, and state the output of the algorithm.

Mark schemes

One mark per valid point unless the scheme says otherwise. Practice questions are written by Comp Sci Wizard in the style of OCR mark schemes.

Question 1 [9 marks] H446/01, June 2022

Levels of response question (9 marks)

AO1: The spiral model has four quadrants (determine objectives, identify and manage risk, develop and test, plan next iteration). Client feedback informs future development and prototypes. Waterfall has structured analysis/design/development/test flow. Progress to next step only after previous step completed.

AO2: Spiral model relies on frequent client feedback, produces functional prototypes where features are added incrementally. More focus on risk; projects may be modified or dropped if risk too great. Waterfall is more structured and reliant on getting requirements correct at start; changes harder to add later. Forces definition to be well understood.

AO3: Spiral involves client feedback, prototypes and evolving projects - better where requirements may change. Waterfall better where requirements are clear and outcomes known. Spiral better for risk management. Waterfall may be more appropriate with large teams due to clearly defined responsibilities.

Question 2 [4 marks] H446/01, June 2019

Extreme programming is a software development methodology
Focus is on good quality code
It is an agile paradigm - designed to allow development to respond to changing user requirements
Involves paired programing
Program is regularly reviewed / iterative process

Suited to this scenario as: Types of virus/threat is continually changing/updating. In order to detect virus effectively there needs to be an emphasis on code quality.

(4 marks - 1 per bullet, max 4)

Question 3 [4 marks] H446/01, June 2017

Prototype is created
(Evaluated and) feedback used to inform next iteration
Any changes are made
Process repeated until... prototype becomes final product

(4 marks - 1 per bullet)

Question 4 [4 marks] Practice question (H446/01 style)

Suitable because:
• Requirements are fixed / well understood, so each stage can be completed in order
• Clear milestones and documentation make progress easy to plan and manage

Drawback (any one, described):
• Testing happens late, so errors found late are expensive to fix
• The customer sees no working software until the end
• Difficult / costly to go back and change requirements once a stage is complete

(4 marks – max 2 for suitability, max 2 for a described drawback)

Question 5 [4 marks] Practice question (H446/01 style)

1 mark for each practice, 1 mark for each explanation, e.g.:
• Pair programming – one programmer writes while the other reviews, so errors are spotted immediately
• Test-driven development / tests written before the code – code must pass the tests, so defects are caught early
• Continuous integration / small frequent releases – problems are found quickly because code is merged and tested often
• Refactoring / simple design – code is kept clean and easy to maintain
• On-site customer – requirements are clarified straight away, so the right thing is built

(4 marks – max 2 practices)

Question 6 [6 marks] Practice question (H446/01 style)

RAD:
• A prototype is built quickly and shown to users / players
• Feedback is used to refine the next iteration
• Suits projects where requirements change

Waterfall:
• Sequential stages, each completed before the next begins
• Requirements are fixed at the start, so changes are difficult / expensive
• No working version until late in the project

Recommendation: RAD, because the requirements will change and player feedback is central to the design (1 mark for a justified recommendation)

(6 marks – max 3 for RAD, max 3 for waterfall, including the recommendation)

Question 7 [3 marks] Practice question (H446/01 style)

• Each loop / iteration begins by identifying and analysing risks
• Risks can be reduced (e.g. by prototyping) before more time and money are committed
• The project can be changed or abandoned at any loop if the risk is too high
• Repeated cycles let the plan be refined as understanding of the problem improves

(3 marks – 1 per point, max 3)

Question 8 [4 marks] Practice question (H446/01 style)

Trace (1 mark per correct row, max 3):
i = 1, total = 1
i = 2, total = 5
i = 3, total = 14
i = 4, total = 30

Output: 30 (1 mark)

(4 marks)