

Week 1

AP Computer Science Principles

Homework bundle for this week

本周作业合集

[exercises.lol](#)

Contents 目录

Topic 1 quiz	2
Exercise sheet 1.1	5
Exercise sheet 1.2	9
Past-paper questions 1.2	13
Exercise sheet 1.3	16
Past-paper questions 1.3	19
Exercise sheet 1.4	20
Past-paper questions 1.4	24

AP Computer Science Principles

Name: _____ Score: _____

1. Collaboration in software means:

- ☐ people working together toward a shared goal
- ☐ one expert doing all the work
- ☐ writing code with no plan
- ☐ avoiding all feedback

2. A program's purpose is why it exists; its function is:

- ☐ what it does — the behavior a user sees
- ☐ how fast it runs
- ☐ who wrote it
- ☐ how much memory it uses

3. Good practice is to start coding immediately, skipping design.

- ☐ True ☐ False

4. The work of finding and fixing mistakes in a program is called _____.

5. Good feedback comments on the work, not the person.

- ☐ True ☐ False

6. Which is an example of input?

- ☐ a user tapping a button
- ☐ text appearing on screen
- ☐ a file being saved
- ☐ a robot arm moving

7. A process that repeats its stages, improving a little each time, is _____.

8. A missing bracket that stops the program from running is a:

- ☐ syntax error
- ☐ logic error
- ☐ runtime error

☐ edge case

9. A _____ control system records every edit and merges everyone's work.

10. A program that waits and reacts to clicks and key presses is _____-driven.

AP Computer Science Principles

1. **people working together toward a shared goal**

A team shares ideas and divides the work toward one goal.

2. **what it does — the behavior a user sees**

Purpose = the goal; function = the visible behavior that reaches it.

3. **False**

Skipping design usually wastes time — plan first, then code.

4. **debugging**

A mistake is a bug; fixing bugs is debugging.

5. **True**

That keeps feedback useful and respectful.

6. **a user tapping a button**

A tap is data the program receives; the others are outputs.

7. **iterative**

Iterative and incremental development loops to refine the program.

8. **syntax error**

Breaking the language's rules is a syntax error.

9. **version**

Version control lets a team combine changes without losing any.

10. **event**

Each event triggers an event handler.

1.1 Collaboration

Name: _____ Class: _____ Date: _____

Total: 9 marks

KEY WORDS users 用户 • pair programming 结对编程 • bias 偏见 • collaboration 协作 • feedback 反馈

Objective

Build the skills to answer exam questions on **collaboration**.

You must be able to:

- explain how **collaboration** 协作 and diverse perspectives improve a solution and reduce **bias** 偏见
- apply **pair programming** 结对编程 (a driver and a navigator)
- describe giving and receiving constructive **feedback** 反馈
- explain why consulting **users** 用户 helps a program meet real needs

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Why collaborate

A team with **diverse perspectives** spots more problems, produces better solutions, and reduces **bias** that one person might miss.

■ Pair programming

Two people share one workstation: the **driver** types the code, and the **navigator** reviews it and suggests direction. They swap roles regularly.

■ Feedback and users

Constructive feedback during development catches issues early; consulting **users and stakeholders** keeps the program aimed at real needs.

2 Practice

2.1 State one benefit of collaboration when developing a program. [1]

2.2 Describe the two roles in pair programming. [2]

2.3 State why consulting users helps a program. [1]

3 Exam-style questions

3.1 In pair programming, the person who types the code is the [1]

- **A** navigator
 - **B** driver
 - **C** tester
 - **D** manager
-

3.2 Diverse perspectives in a team help to reduce [1]

- **A** speed
 - **B** bias
 - **C** cost
 - **D** memory use
-

3.3 A team builds an app without asking any users for input.

(a) State one risk of doing this. [1]

(b) State one collaboration practice that would help. [1]

(c) State one benefit of that practice.

[1]

Solutions

2.1 any one of: better solutions, fewer errors, reduced bias, more ideas.

2.2 the driver types the code; the navigator reviews it and guides the direction.

2.3 it keeps the program aimed at users' real needs.

3.1 B.

3.2 B.

3.3 (a) the app may not meet users' real needs. (b) consult users, or use pair programming (any one). (c) fewer errors, or a program that better fits its purpose.

1.2 Program Function and Purpose

Name: _____ Class: _____ Date: _____

Total: 10 marks

KEY WORDS input 输入 • output 输出 • purpose 目的 • event 事件

Objective

Build the skills to answer exam questions on **program function and purpose**.

You must be able to:

- explain the **purpose** 目的 of a program (the problem it solves)
- describe how a program **functions** as a sequence of instructions
- identify a program's **inputs** 输入 and **outputs** 输出
- explain how an **event** 事件 affects execution

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Purpose vs function

The **purpose** is the problem the program solves or the value it provides; the **function** is **how** it works — the sequence of instructions it runs.

■ Inputs and outputs

Inputs can be text, clicks, images, audio, or data from another program; **outputs** can be displayed text, sound, visuals, or tactile feedback.

■ Events

An **event** is an action associated with input (e.g. a click or key press) that triggers part of the program to run.

2 Practice

2.1 State the difference between a program's purpose and its function. [2]

2.2 Give two examples of program input. [2]

2.3 Define an event in a program. [1]

3 Exam-style questions

3.1 The purpose of a program is [1]

- **A** the language it is written in
- **B** the problem it solves
- **C** its file size
- **D** its author

3.2 A mouse click that starts an action is an example of an [1]

- **A** output
- **B** event
- **C** variable
- **D** loop

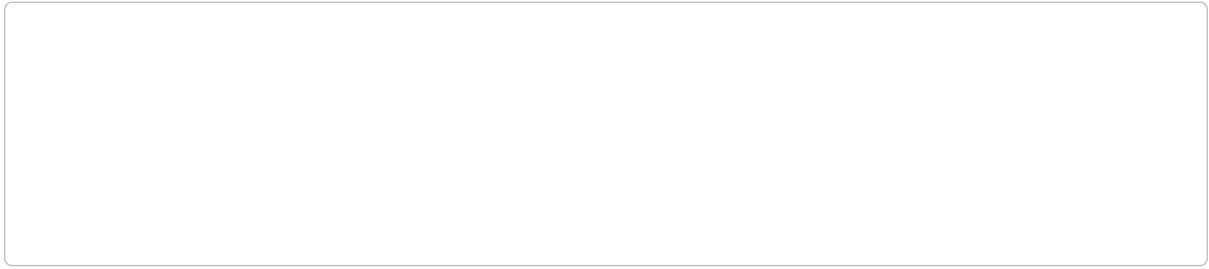
3.3 A weather app takes a city name and shows the forecast.

(a) State its input. [1]

(b) State its output. [1]

(c) State its purpose.

[1]



Solutions

2.1 the purpose is the problem it solves; the function is how it works (its sequence of instructions).

2.2 any two of: text typed, a mouse click, an image, audio, sensor data.

2.3 an action associated with input that triggers part of the program to run.

3.1 B.

3.2 B.

3.3 (a) the city name. (b) the displayed forecast. (c) to give the user weather information.

1. Describe one piece of documentation that would be appropriate to include with or in your program. Explain how this documentation can be used by another programmer to better understand a particular code segment in your program.

1. Identify an example output of your program. Explain how this output shows an aspect of your program's functionality.

COMPUTER SCIENCE PRINCIPLES**SECTION II****Time—1 hour****2 Questions**

Directions: Read each question carefully and completely. Write your response in the space provided for each question in the Written Response booklet.

You may plan your answers in this orange booklet, but no credit will be given for anything written in this booklet. **You will only earn credit for what you write in the separate Written Response booklet.**

1. Programs accept input to achieve their intended functionality. Describe at least one valid input to your program and what your program does with that input.

Write your responses to this question only on the designated pages in the separate Written Response booklet. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE PRINCIPLES**SECTION II****Time—1 hour****2 Questions**

Directions: Read each question carefully and completely. Write your response in the space provided for each question in the Written Response booklet.

You may plan your answers in this orange booklet, but no credit will be given for anything written in this booklet. **You will only earn credit for what you write in the separate Written Response booklet.**

1. Identify the expected group of users of your program. Explain how your program addresses at least one concern or interest of the users you identified.

Write your responses to this question only on the designated pages in the separate Written Response booklet. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

1.3 Program Design and Development

Name: _____ Class: _____ Date: _____

Total: 8 marks

KEY WORDS comments 注释 • iterative 迭代 • program requirements 程序需求 • prototype 原型

Objective

Build the skills to answer exam questions on **program design and development**.

You must be able to:

- describe the **iterative** 迭代 and incremental nature of development
- use a process to investigate, design, **prototype** 原型, and test
- create a **program requirements** 程序需求 description
- document a program with clear **comments** 注释

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Iterative development

Programs are built in **repeated cycles**: design → prototype → test → refine, again and again, improving each time.

■ Requirements and planning

A **requirements** description captures what the user needs. Good practice is to **plan** the program before writing code.

■ Comments

Comments are notes in the source code that explain **how and why** it works, helping others (and your future self) understand it.

2 Practice

2.1 State what “iterative” development means.

[1]

2.2 State the purpose of a program requirements description.

[1]

2.3 State why programmers add comments to their code. [1]

3 Exam-style questions

3.1 An iterative development process is one that [1]

- **A** never changes
 - **B** repeats and refines
 - **C** skips testing
 - **D** has no plan
-

3.2 Comments in a program are used to [1]

- **A** make it run faster
 - **B** explain how and why the code works
 - **C** store data
 - **D** create loops
-

3.3 A student plans, builds, tests, then improves a program over several cycles.

(a) Name this kind of process. [1]

(b) State one step done before writing code. [1]

(c) State one benefit of adding comments. [1]

Solutions

2.1 developing in repeated cycles, refining the program each time.

2.2 to capture what the user needs the program to do.

2.3 to explain how and why the code works.

3.1 B.

3.2 B.

3.3 (a) an iterative (and incremental) process. (b) investigate needs / write requirements / design or plan (any one). (c) it makes the program easier to understand and maintain.

2. Refer to your Personalized Project Reference when answering this question.

- A. Consider the first selection statement included in the Procedure section of your Personalized Project Reference. Identify the Boolean expression in this selection statement. Identify a specific value or set of values that will cause this expression to evaluate to `false`. Explain why the specified value(s) will cause this expression to evaluate to `false`.
- B. Consider the code segment in part (ii) of the List section of your Personalized Project Reference. Suppose another programmer modifies this code segment. Describe a modification the other programmer could make to this code segment that would result in a logic error. Explain why this modification would result in a logic error.
- C. Consider the procedure identified in part (i) of the Procedure section of your Personalized Project Reference. Describe the functionality provided by this procedure. Explain how implementing this functionality as a procedure results in your program being easier to maintain than if the functionality were not implemented as a procedure.

STOP

END OF EXAM

1.4 Identifying and Correcting Errors

Name: _____ Class: _____ Date: _____

Total: 9 marks

KEY WORDS logic error 逻辑错误 • syntax error 语法错误 • edge cases 边界情况 • testing 测试 • runtime error 运行时错误

Objective

Build the skills to answer exam questions on **identifying and correcting errors**.

You must be able to:

- distinguish a **logic error** 逻辑错误, a **syntax error** 语法错误, a runtime error, and an overflow error
- use **testing** 测试 with varied inputs, including **edge cases** 边界情况
- locate and fix a defect during **debugging** 调试
- verify the corrected program produces the intended output

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Types of error

- **Syntax error** — breaks the language's grammar, so it will not run.
- **Logic error** — runs but gives the **wrong** result.
- **Runtime error** — crashes while running (e.g. dividing by zero).
- **Overflow error** — a value is too large for the bits available.

■ Testing and debugging

Test with varied inputs, including **edge cases** (boundaries and unusual values). To find a bug, **hand-trace** the code or add temporary **print statements**, then re-test to verify the fix.

2 Practice

2.1 State the difference between a syntax error and a logic error. [2]

2.2 State what edge cases are used for in testing. [1]

2.3 Name one debugging strategy. [1]

3 Exam-style questions

3.1 A program runs but produces the wrong answer. This is a [1]

- **A** syntax error
- **B** logic error
- **C** overflow error
- **D** case with no error

3.2 Testing a program with unusual or boundary inputs uses [1]

- **A** edge cases
- **B** comments
- **C** variables
- **D** loops

3.3 A program crashes when it tries to divide by zero.

(a) Name the type of error. [1]

(b) Name one way to find the bug. [1]

(c) State what you should check after fixing it.

[1]

Solutions

2.1 a syntax error breaks the language's rules so the program will not run; a logic error runs but gives the wrong result.

2.2 to test the program at boundaries and unusual values where defects often hide.

2.3 hand-tracing the code, or adding print statements / using a debugger (any one).

3.1 B.

3.2 A.

3.3 (a) a runtime error. (b) hand-trace or add print statements to find where it fails (any one). (c) that it now produces the intended output for its requirements.

2. Refer to your Personalized Project Reference when answering this question.

- A. Consider the first iteration statement in the Procedure section of your Personalized Project Reference. Identify the variable(s) that determine when the iteration statement stops iterating. Identify specific values for the variable(s) that will cause the iteration statement to stop. Explain why these value(s) would cause the iteration statement to stop. If there are no such variable(s) or specific value(s), explain how the iteration statement stops.
- B. Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Another programmer is planning to use your procedure in a different program and wants to write some test cases to check whether it will work. Write a procedure call with specific argument(s) that your procedure will accept but will cause your procedure to behave incorrectly. Describe the incorrect behavior. Explain why the incorrect behavior occurs as a result of this call. If it is not possible for a call with accepted arguments to cause your procedure to behave incorrectly, explain why this is the case for your procedure.
- C. Consider the list included in part (i) of the List section of your Personalized Project Reference. Explain how the list uses abstraction to manage complexity in your program. Suppose the list was not included in your program. Describe how you would need to adjust the code segment in part (ii) of the List section of your Personalized Project Reference so that it has the same behavior. If it is not possible to have the same behavior without the use of the list, explain why.

STOP
END OF EXAM

2. Refer to your Personalized Project Reference when answering this question.

- A. Consider the first selection statement included in the Procedure section of your Personalized Project Reference. Identify the Boolean expression in this selection statement. Identify a specific value or set of values that will cause this expression to evaluate to `true`. Explain why the specified value(s) will cause the expression to evaluate to `true`.
- B. Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Suppose another programmer modifies the code within this procedure. Describe a modification the other programmer could make that would cause this procedure to have a logic error. Describe how the behavior of this procedure would change because of the error.
- C. Consider the list included in the List section of your Personalized Project Reference. Suppose another programmer adds several new elements to the end of the list. Explain how the code segment in part (ii) of the List section would need to be modified to account for the additional elements. If no changes to the code segment are necessary, explain why this is the case for your program.

STOP

END OF EXAM

1. Identify an unexpected or invalid input that a user could provide to your program. Describe the behavior of your program after it receives this input. If it is not possible for your program to accept an unexpected or invalid input, explain why this is the case.

2. Refer to your Personalized Project Reference when answering this question.

- A. Consider the first selection statement included in the Procedure section of your Personalized Project Reference. Identify the Boolean expression in this selection statement. Identify a specific value or set of values that will cause this expression to evaluate to `false`. Explain why the specified value(s) will cause this expression to evaluate to `false`.
- B. Consider the code segment in part (ii) of the List section of your Personalized Project Reference. Suppose another programmer modifies this code segment. Describe a modification the other programmer could make to this code segment that would result in a logic error. Explain why this modification would result in a logic error.
- C. Consider the procedure identified in part (i) of the Procedure section of your Personalized Project Reference. Describe the functionality provided by this procedure. Explain how implementing this functionality as a procedure results in your program being easier to maintain than if the functionality were not implemented as a procedure.

STOP

END OF EXAM

