Desk Number _____

Student Number _____

Student Name _____

School of Science and Engineering

MIDTERM EXAMINATION

Term 2, 2025

CSC1001 Introduction to Computer Science

Examination Duration: 120 minutes

Reading Time: 10 minutes

This examination has __5__ questions.

The score distribution of the 5 question is Question 1(40%), Question 2(20%), Question 3(10%), Question 4(15%), Question 5(15%)

Exam Conditions:

This is a FORMAL Examination

This is a RESTRICTED OPEN BOOK Exam. Maximum of one (1) sheet of handwritten notes double-sided are permitted

Materials Permitted In The Exam Venue:

Maximum of one (1) sheet of handwritten notes double sided are permitted. **NO OTHER MATERIALS PERMITTED**

Calculators without the functionalities of programming and file storage are permitted.

Materials To Be Supplied To Students:

1 × 6 Page Answer Booklet

You must write all your answers in your answer book.

Question 1. (40%)

Please describe what will be printed for each program as below.

	Program	Printed Results
1) 3%	`x = (2**3 % 5) // 2 + float("77".replace("7", "1"))` `print(x)`	
2) 3%	`s1, s2, s3 = eval("15-3"), str(15-3), "15-3"` `print(s1 + len(s2) + len(s3))`	
3) 3%	`t = [10, 20, "python", False, 80]` `print(t[2][2:5].upper(), t[0:2])`	

4) 3%	`x = 10`
	`y = []`
	
	`if (not x > 5) or (y and x % 2 == 0):`
	`  print("csc1001")`
	`elif len(y) == 0 and x > 8:`
	`  print("hello")`
	`else:`
	`  print("world")`	

5) 3%	`n = 12`
	`i = 1`
	`while i < n:`
	`  print(i, end="*")`
	`  i = i * 2`	

6) 4%	`def log_event(event, cache=[], max_size=1):`
	`  cache.append(event)`
	`  if len(cache) > max_size:`
	`  cache[:] = cache[1:]`
	`  print(cache)`
	
	`log_event("start")`
	`log_event("finish", max_size=2)`	

7) 4%	`data = "student_ID_12345.csv"`
	`course = ""`
	`recording = False`
	`for char in data:`
	`  if char == '_':`
	`  recording = not recording`
	`  continue`
	`  if char == '.':`
	`  break`
	`  if recording:`
	`  course += char`
	`  print(course)`	

8) 3%	`for i in range(3):`
	`  if i == 2:`
	`  print(i, end=",")`
	`  break`
	`  else:`
	`  print("Loop finished")`	

9) 3%	`s = "user-tongxin-li-admin@example.com"` `print(s.split("@")[0].split("-")[-1])`	

10) 4%	`counts = {'a': 5, 'b': 10}`
	`counts['c'] = counts.get('c', 0) + 1`
	`print(counts['a'], counts['c'])`
	`print('b' in counts)`	

11) 3%	`h = open('example.txt', 'w')`
	`h.write('Hello')`
	`h.close()`
	`h = open('example.txt', 'r')`
	`s = h.read()`
	`print('First read:', s)`
	`s = h.read()`
	`print('Second read:', s)`	

12) 4%	`d = {'b':1, 'a':2, 'c':0}`
	`lst = []`
	`for k, v in d.items():`
	`  if v > 0:`
	`  lst.append(k)`
	`  lst.sort()`
	`  print(lst)`	

Question 2. (5*4% = 20%)

Please answer the following short questions.

1) Provide the Binary and Hexadecimal representations of the decimal number 29.25.

2) Please list two advantages of using binary numbers in modern computer systems.

3) Please describe the differences between arguments and parameters.

4) Will the following be implemented without causing an error? (Yes/No)

```
data = ("apple", 5, "orange")
print(data[0])
data[1] = 10
print(data)
```

Explain the difference between a list and a tuple in Python. Provide one reason to use a tuple instead of a list.

5) What is the purpose of the return keyword in a function? What happens if a function does not have a return statement?

Question 3. (10%)

Write a Python program that plays a number-guessing game with a user. The game should follow these rules:

- The computer fixes a **secret random integer** between 1 and 50.
- The user gets a maximum of **5** attempts to guess the number.
- After each guess, the program must provide feedback to the user, telling them if their guess was "**Too high**!" or "**Too low!**".
- If the user guesses correctly, the game ends, and the program prints a success message, like "**Correct! You guessed it in [N] attempts.**" where N is the total number of attempts.
- If the user runs out of attempts, the game ends, and the program should reveal the secret number, printing "**Sorry, you've run out of attempts. The secret number was [X].**" where X is the correct integer.

Question 4. (15%)

Please first explain the usage of the built-in string method **find()**. Then, write a function **myFind**(*main_string*, *sub_string*):

- *main_string* is the string to be searched.
- *sub_string* is the string to search for.
- The function should return the **ending index** of the first occurrence of *sub_string* within *main_string*.

- If *sub_string* is not found, the function should return -1.

You may not use the built-in find() or index() methods in your implementation.

(e.g., if *main_string* is "hello world csc1001" and *sub_string* is "world", the result should be 10)

Question 5. (15%)

You are in charge of counting the votes for a class election. You have a file, **votes.txt**, where every line contains the name of the candidate who received a vote.

Example **votes.txt** content:
Alice
Bob
Alice
Charlie
Alice
Bob
…

Please write a Python program that:

- Reads the **votes.txt** file.
- Counts the total number of votes for each candidate.
- Creates a new file named **results.txt** that lists each candidate and their total vote count, sorted alphabetically by the candidate's name.

END OF EXAMINATION