

CONFIDENTIAL



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**FINAL EXAMINATION  
SEMESTER II  
SESSION 2023/2024**

COURSE NAME : PROGRAMMING PRINCIPLES  
COURSE CODE : DAT10603  
PROGRAMME CODE : DAT  
EXAMINATION DATE : JULY 2024  
DURATION : 3 HOURS  
INSTRUCTIONS :  
1. ANSWER ALL QUESTIONS  
2. THIS FINAL EXAMINATION IS CONDUCTED VIA  
 Open book  
 Closed book  
3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSED BOOK

THIS QUESTION PAPER CONSISTS OF **EIGHT (8)** PAGES

CONFIDENTIAL

**TERBUKA**

**PART A (20 MARKS)**

Answer **True** or **False** for each of the following questions.

- Q1** A variable must be defined before it can be used.
- Q2** The `cout` object is used with the stream extraction operator.
- Q3** `x >= y` is the same as `(x > y || x == y)`.
- Q4** The `=` operator and the `==` operator perform the same operation when used in a Boolean expression.
- Q5** A named constant with the `const` keyword must be initialised during definition.
- Q6** When an `if` statement is nested in another `if` statement, the only time the inner `if` is executed is when the Boolean expression of the outer `if` is false.
- Q7** If a `switch` block does not have `break` statements, the program executes all the lines from the matching `case` statement to the end of the block.
- Q8** It is not necessary to initialise a loop's counter variable.
- Q9** In a nested loop, the inner loop goes through all its iterations for every single iteration of the outer loop.
- Q10** When a `char` array is partially initialised, the uninitialized elements will be set to the null character (`\0`).
- Q11** The index of an array may be a floating-point number.
- Q12** The `toupper('A')` function call will return the character A.
- Q13** The `strcpy` function appends the second C-string to the first C-string.
- Q14** Function prototypes are terminated with a semicolon.
- Q15** When a function terminates, it always branches back to the function it was called from.
- Q16** The `&` operator is used to get the address of a variable.
- Q17** When the indirection operator is used with a pointer variable, we are working with the value the pointer is pointing to.
- Q18** A pointer variable can be used as a function parameter.
- Q19** When you call an `ofstream` object's `open` member function, the specified file will be overwritten or erased if it already exists.
- Q20** The stream insertion operator can be used to write data to files.

(20 marks)

**PART B (30 MARKS)**

**Q21** The C++ basic character set consists of four categories. Classify the following characters into their appropriate character set.

(a) z

(1 mark)

(b) 9

(1 mark)

(c) !

(1 mark)

**Q22** There are five types of tokens in C++. Classify the following tokens into their appropriate type.

(a) using

(1 mark)

(b) totalCharge

(1 mark)

(c) 3.142

(1 mark)

(d) &&

(1 mark)

(e) #

(1 mark)

**Q23** C++ provides various data types such as int, float, double and char. Suggest a suitable data type to store the following literal constant.

(a) 9

(1 mark)

(b) -9.807

(1 mark)

(c) 'A'

(1 mark)

(d) '9'

(1 mark)

(e) "I love Principles of Programming" (1 mark)

(f) "A" (1 mark)

**Q24** Determine whether the following variables are **valid** or **invalid** according to C++ naming rules.

(a) heatcapacity (1 mark)

(b) TOT4L (1 mark)

(c) x (1 mark)

(d) 5fingers (1 mark)

**Q25** The `iomanip` library allows us to perform output formatting. Identify the output of the program in **Figure Q25.1**.

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     const double PI = 3.142;
7     double num = 10;
8
9     cout << setprecision(4);
10    cout << PI << "\t" << num << endl;
11
12    cout << setprecision(4) << showpoint;
13    cout << PI << "\t" << num << endl;
14
15    cout << setprecision(4) << fixed;
16    cout << PI << "\t" << num << endl;
17
18    return 0;
19 }
```

**Figure Q25.1** Formatting Output

(6 marks)

- Q26** Type casting allows us to manually perform data type conversion. Identify the output of the program in **Figure Q26.1**.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int dividend = 7;
6     int divisor = 2;
7
8     cout << dividend / divisor << endl;
9     cout << static_cast<float>(dividend) / divisor << endl;
10    cout << dividend / static_cast<int>(divisor) << endl;
11
12    return 0;
13 }
```

**Figure Q26.1** Type Casting

(3 marks)

- Q27** Given the incomplete program in **Figure Q27.1**, where lines 10, 16 and 17 are missing the necessary statement for the program to execute correctly.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     string name, university;
7     int age;
8
9     cout << "Enter full name: ";
10    // ____Q27 (a)____
11
12    cout << "Enter age: ";
13    cin >> age;
14
15    cout << "Enter full university name: ";
16    // ____Q27 (b)____
17    // ____Q27 (c)____
18
19    cout << name << " is a ";
20    cout << age << " years old student, studying at ";
21    cout << university << "." << endl;
22
23    return 0;
24 }
```

**Figure Q27.1** Incomplete Program

- (a) Write a statement that should be placed at line 10.
- (b) Write a statement that should be placed at line 16.
- (c) Write a statement that should be placed at line 17.

(3 marks)

**PART C (50 MARKS)**

**Q28** Write a program that asks the user to enter a text. The program should then list all the alphabets in that text, separated with a space. **Figure Q28.1** is the program output, with example input shown in bold.

```

Enter text: Contact CeDS UTHM at +60 6-974 2187 (Sun-Thu, 0900-1600).

The alphabets in the text:
C o n t a c t C e D S U T H M a t S u n T h u
    
```

**Figure Q28.1** List Alphabets

(8 marks)

**Q29** Students of Diploma in Information Technology will be organising an Information Technology Quiz. Five schools around Muar will be sending their contestant to this quiz. The quiz consists of two rounds, where each contestant will sit through both rounds. Write a program that performs the following tasks:

- Ask the user to enter the score each contestant receives in each round, and store it in an array named `score[5][2]`. Use nested loop.
- Calculate and display the average score each contestant receives from both rounds. Use another nested loop. Make sure to format the output accordingly.

**Figure Q29.1** is the program output, with example input shown in bold.

```

ROUND 1
Enter score [0 to 20] for contestant 1: 20
Enter score [0 to 20] for contestant 2: 19
Enter score [0 to 20] for contestant 3: 10
Enter score [0 to 20] for contestant 4: 16
Enter score [0 to 20] for contestant 5: 18

ROUND 2
Enter score [0 to 20] for contestant 1: 18
Enter score [0 to 20] for contestant 2: 15
Enter score [0 to 20] for contestant 3: 12
Enter score [0 to 20] for contestant 4: 19
Enter score [0 to 20] for contestant 5: 15

-----
Contestant      Average Score
-----
1                19.0
2                17.0
3                11.0
4                17.5
5                16.5
-----
    
```

**Figure Q29.1** Quiz Score

(20 marks)



**Q30** Write a program that calculates the number of bad calories in a recipe. Calories are considered bad when coming from fat and sugar. The program must have all the functions listed in **Table Q30.1**.

**Table Q30.1** Purpose of Each Function

Name	Purpose
getWeight	<ul style="list-style-type: none"> <li>• Argument: ingredient name. The value of the argument should either be <i>fat</i> or <i>sugar</i>.</li> <li>• Then, ask the user to enter weight in kilograms. A prompt using the value of the argument should be displayed before the cin object.</li> <li>• Input validation: Do not accept weight less than 0.</li> <li>• Return value: weight.</li> </ul>
calcCaloriesFat	<ul style="list-style-type: none"> <li>• Argument: fat weight.</li> <li>• Then, calculate the amount of calories by multiplying the weight by 9.</li> <li>• Return value: fat calories.</li> </ul>
calcCaloriesSugar	<ul style="list-style-type: none"> <li>• Argument: sugar weight.</li> <li>• Then, calculate the amount of calories by multiplying the weight by 4.</li> <li>• Return value: sugar calories.</li> </ul>
main	<ul style="list-style-type: none"> <li>• Call the getWeight function to get the fat weight.</li> <li>• Call the getWeight function to get the sugar weight.</li> <li>• Call the calcCaloriesFat to calculate the fat calories.</li> <li>• Call the calcCaloriesSugar to calculate the sugar calories.</li> <li>• Finally, calculate and display the total of bad calories.</li> </ul>

**Figure Q30.1** and **Figure Q30.2** are the program output, with example input shown in bold.

```

Enter fat weight (kilograms): -0.5
ERROR: Weight must be a positive value
      Re-enter fat weight (kilograms): -5
ERROR: Weight must be a positive value
      Re-enter fat weight (kilograms): 0.5
Enter sugar weight (kilograms): 0.3
Total of bad calories: 5.7
    
```

**Figure Q30.1** Bad Calories Example Input A



Enter fat weight (kilograms): <b>0.5</b> Enter sugar weight (kilograms): <b>0.3</b> Total of bad calories: 5.7
--

**Figure Q30.2** Bad Calories Example Input B

(22 marks)

**- END OF QUESTIONS -**

**TERBUKA**