CONFIDENTIAL

UNIVERSITI TEKNOLOGI MALAYSIA
FACULTY OF COMPUTING

FINAL EXAMINATION (WRITTEN)

SEMESTER I 2022/ 2023

SUBJECT CODE : **SECJ1013**
SUBJECT NAME : **PROGRAMMING TECHNIQUE I**
YEAR/COURSE : **1 (SECJH/SECVH/SECBH/SECRH/SECPH)**
TIME : **2 Hours**
DATE :
VENUE :

INSTRUCTIONS TO THE STUDENTS:

This exam book consists of five (5) questions.

ANSWER ALL THE QUESTIONS AND WRITE THE ANSWERS IN THE ANSWER BOOKLET.

Name	
Matric No.	
Section	
Lecturer's Name	

*(This question booklet consists of **8 pages** INCLUDING this page)*

QUESTION 1 **[20 MARKS]**

a) Write a function declaration (**function prototype**) and a **function definition** for the **`void`** function **`calcSumAvg`**, which computes the sum and average of two integer numbers. The function should have four parameters: two **`int`**s, an **`int`** reference, and a **`float`** reference. The **`int`** parameters should have a default argument, which are 3 and 12. The parameters are not necessarily in the order listed. Finally, write an example **call to the function**.

(8 marks)

b) Assume that the following program segments run without errors. Trace each program and write the output.

i) (5 marks)

```
1   char str[] = "@T1fINal";
2   for (int x = 0; x < strlen(str); x++) {
3     if (x < 3) {
4        if (isalpha(str[x]))
5           cout << "you " << str[x] << endl;
6        else
7        if (isdigit(str[x]))
8           cout << "you rock" << endl;
9        else
10          cout << "count on you" << endl;
11    }
12
13    if (isupper(str[x]))
14       str[x] = '*';
15    else
16    if (x % 2)
17       str[x] = str[x-1];
18  }
19  cout << str << endl;
```

ii) (7 marks)

```
1   int m = 5;
2
3   void function1(float &a, int &b) {
4     static int c = 1000;
5     a *= m;
6     cout << a << " " << b++ << " " << --c << endl;
7   }
8
9   int function2(float &y, int x = 200) {
10    int z = 5;
11    x /= y;
12    z += y;
13    cout << x << " " << y-- << " " << z << endl;
```

```
14    return x + y;
15  }
16
17  int main() {
18    int p = 100;
19    float q = 2.5;
20    function1(q, m);
21    p = function2(q);
22    function1(q, p);
23    cout << function2(q, m) << " " << p << endl;
24    return 0;
25  }
```

QUESTION 2 **[20 MARKS]**

a) Given the following array definition:

```
float marks[100][6];
```

i) How many columns does the array have? (0.5 mark)

ii) How many elements does the array have? (0.5 mark)

iii) Write a C++ statement that stores the value 89.9 in the last column of the last row of the array. (1 mark)

b) Given the array initialization below, write C++ statements for the following questions:

```
const int NUM = 7;
char name[][10] = {"dafi", "alif", "amri", "nawal", "azhar",
"nazif", "karim"};
```

i) Using a loop, display all seven names in the array. The statements should produce the output shown in **Figure 1**. (3.5 marks)

```
name[0] = dafi
name[1] = alif
name[2] = amri
name[3] = nawal
name[4] = azhar
name[5] = nazif
name[6] = karim
```

Figure 1: Sample output for Question 2b(i)

ii) Using a loop, display only the name(s) in the array that begin with the letter 'a' and change the letter to uppercase. The statements should produce the output shown in **Figure 2**. (6.5 marks)

```
Alif
Amri
Azhar
```

Figure 2: Sample output for Question 2b(ii)

c) Assume that the following program segment runs without errors. Trace the program and write its output. (8 marks)

```
1  int i, j;
2  int a[3][3] = {99, 77, 22, 44, 55, 33, 66, 88, 11};
3
4  for(i = 1; i < 3; i = i + 1)
5     for(j = i; j >= 1; j = j - 1)
6        cout << i << " " << j << " "
7             << ++a[i-1][2-j] << endl;
8  cout << i << " " << j << " " << a[i-2][j+2] << endl;
```

QUESTION 3 **[20 MARKS]**

a) Locate and identify the syntax and/or logical errors in **Program 1**. Specify the error and the line number where it occurred. (5 marks)

```
1  //Program 1
2  #include <iostream>;
3  using namespace std;
4
5  int main
6  {
7    float number, half;
8    cout << "Enter a number and I will divide it ";
9    cout << "in half for you.\n";
10   cin >> numberr;
11   half = half / 2;
12   cout << fixedpoint << showpoint << half;
13   return 0;
14 }
```

b) The stream manipulator is used in the following program segment. Assume that the program segment runs without errors. If the program's input is "**C++ Program**", trace the program and write its output. ***Note:*** Use '#' to represent one whitespace. (5 marks)

```
1  float data = 15.678394;
2  char s[8];
3
4  cin  >> setw(9) >> s;
```

```
5   cout << setw(5) << s << endl;
6   cout << setprecision(6) << data << endl;
7   cout << setprecision(4) << static_cast<int>(data)/2 << endl;
8   cout << setw(6) << setprecision(3) << data * 5 << endl;
9   cout << fixed << data << endl;
```

c) Complete **Program 2** with appropriate C++ statements based on the comments provided (**bold** text). (10 marks)

```
1   //Program 2
2   //(i) Include a suitable library for file operations (1 mark)
3   _____________________________________________________________
4   #include <iostream>
5   using namespace std;
6
7   int main () {
8     int num;
9     //(ii) Open the input file named "input.txt" (2 marks)
10    _____________________________________________________________
11
12    //(iii) Check the file has opened successfully (1.5 marks)
13    _____________________________________________________________
14    {
15       cout << "Input file could not be opened!\n";
16       return 0;
17    }
18
19    //(iv) Read the number from the input file (1 mark)
20    _____________________________________________________________
21
22    //(v) Continue reading until you reach the end of the
23    //input file (1.5 marks)
24    _____________________________________________________________
25    {
26       //(vi) Display the square of the number on the
27       //screen (1 mark)
28       _____________________________________________________________
29
30       //(vii) Read the number from the input file again
31       //(1 mark)
32       _____________________________________________________________
33    }
34
35    //(viii) Close the input file (1 mark)
36    _____________________________________________________________
37    return 0;
38  }
```

QUESTION 4 **[20 MARKS]**

a) Draw the memory layout for each of the following C++ statements:

i) `float *fltPtr;` (1 mark)

ii) `float num = 3.5;` (1 mark)

iii)
```
char ch = '&';
char *charPtr = &ch;
```
(2 marks)

b) Assume that the following program segment runs without errors. Trace the program and write its output. Note that, the memory addresses for the variables are as follows: **`ch = 0x6ffe1c`**, **`smallNum = 0x6ffe08`**, **`anotherSmallNum = 0x6ffe04`**, **`charPtr = 0x6ffdf8`**, **`numPtr = 0x6ffdf0`**, and **`smlNumPtr = 0x6ffde8`**. (6 marks)

```
1   char ch = '^';
2   int  smallNum = 6;
3   int  anotherSmallNum = 3;
4
5   char *charPtr = &ch;
6   int  *numPtr = &smallNum;
7   int  *smlNumPtr = &anotherSmallNum;
8
9   cout << *charPtr << endl;
10  cout << numPtr << endl;
11
12  cout << (*numPtr)++ + --(*smlNumPtr) << endl;
13
14  cout << *numPtr << endl;
15  if (*numPtr > *smlNumPtr)
16        numPtr = &anotherSmallNum;
17
18  cout << *numPtr << endl;
19  cout << *smlNumPtr << endl;
```

c) Complete **Program 3** with appropriate C++ statements based on the comments provided (**bold** text). Please keep in mind that you may need to write multiple C++ statements to answer some of the questions in the comments. (10 marks)

```
1   //Program 3
2   #include <iostream>
3   #define N 5
4   using namespace std;
5
6   int main()
7   {
8      int *numPtr;
9      float *fltPtr;
```

```
10
11      //(i) Dynamically allocate memory to the integer pointer
12      //variable, numPtr (1 mark)
13      _____________________________________________________________
14
15      //(ii) Dynamically allocate an array of size N to the float
16      //pointer variable, fltPtr (2 marks)
17      _____________________________________________________________
18
19      cout << "Enter 5 numbers: ";
20      for (int cnt = 0; cnt < N; cnt+=1)
21      //(iii) Read five numbers entered by the user using pointer
22      //and insert it into the dynamic array, fltPtr (2 marks)
23      _____________________________________________________________
24
25      //(iv) Assign a value of 5 to the dynamic integer, numPtr
26      //(1 mark)
27      _____________________________________________________________
28
29      //(v) Update the value of the even-indexed element in the
30      //dynamic array, fltPtr, by adding its current value to
31      //the value pointed to by the numPtr pointer (2 marks)
32      for (int cnt = 0; cnt < N; cnt+=2)
33        _____________________________________________________________
34
35      //(vi) Free all dynamically allocated memory (2 marks)
36      _____________________________________________________________
37
38      return 0;
39  }
```

QUESTION 5 **[20 MARKS]**

a) Write C++ statements to answer the following questions:

i) Define a structure called **`Options`** with four members. One integer **`Nit`** with a value of 100, two floating point numbers **`learningRate`** and **`error`** with values of 0.025 and 1e-5, respectively, and a character variable **`opt`**. (4 marks)

ii) Declare the structure in 5a(i). (1 mark)

iii) Using the declared structure in 5a(ii), assign the character '**`y`**' to structure's member **`opt`**, then print the assigned character and the value of **`Nit`** on the screen. (3 marks)

b) You are given the C++ code segment below to access structure and nested structure members. The structure member **`functionName`** stores an array of characters, "**`Rosenbrock`**". While, the nested structure member **`functionType`** stores an array of

characters, "**Convex**". With this information, define the two structures that will allow you to use the code below, and then write the output of the two **cout** statements in the code. (7 marks)

```
1   Algorithm alg;
2
3   cout << alg.functionName << endl;
4   cout << alg.fnc.functionType[5] << endl;
```

d) Assume that the program segment runs without errors. Trace the program and write its output. (5 marks)

```
1   enum Algorithm {ADMM, AMA, ADAM, AM, RMSPROP};
2
3   Algorithm algo1 = AMA;
4   Algorithm algo2 = ADAM;
5   Algorithm algo3 = RMSPROP;
6   int add, num;
7
8   cout << algo1 << endl;
9   cout << algo2 << endl;
10
11  add = algo1 + algo2;
12  if (add < algo3) {
13     algo1 = AM;
14     num = algo3 + algo1;
15     algo2 = static_cast<Algorithm>(num);
16  }
17
18  cout << algo3 << endl;
19  cout << algo2 << endl;
20  cout << algo1 << endl;
```