EXAMINATION COVERSHEET

Autumn 2023 Final Examination

UNIVERSITY
OF WOLLONGONG
IN DUBAI

THIS EXAMINATION CONTENT IS STRICTLY CONFIDENTIAL
Students must comply with requirements stated in the Examination Policy & Procedures

Student Number:	
First Name:	
Family Name:	
Date of Examination: (DD/MM/YY)	**/12/2023**
Subject Code:	**CSCI291**
Subject Title:	**Programming for Engineers**
Time Permitted to Write Exam:	2 Hours
Total Number of Questions:	**5**
Total Number of Pages *(including this page):*	**4**

INSTRUCTIONS TO STUDENTS FOR THE EXAM

1. Please note that subject lecturer/tutor will be unavailable during exams. *If there is a doubt in any of the exam questions i.e. problem solving etc. students should proceed by assuming values etc. Students should mention their assumption on the question paper.*
2. Answers must be submitted in Moodle.
3. Answer ALL questions. The marks for each question are shown next to each question.
4. Total marks: 45. This Exam is worth 45% of your final marks for CSCI291.

EXAMINATION MATERIALS/AIDS ALLOWED
Appendix and approved calculator

Exam Unauthorised Items - Students bringing these items to the examination room must follow the instructions of the invigilators with regards to these items.

5. Bags, including carrier bags, backpacks, shoulder bags and briefcases.
6. Any form of electronic device including but not limited to mobile phones, smart watches, MP3 players, handheld computers and unauthorised calculators.
7. Calculator cases and covers, opaque pencil cases.
8. Blank paper
9. Any written material

NOTE: The University does not guarantee the safe-keeping of students' personal items during examinations. Students concerned about the safety of their valuable items should make alternative arrangements for their care.

IMPORTANT:

- **Unless it is a set requirement, a user defined function should NOT include a printf/cout or scanf/cin statement.**
- **If asked to write the code of a function, write its code only without the main() function, program headers and user defined function prototypes.**
- **If asked to write the full C program include the main() function, the program headers and user defined function prototypes.**
- **In the following, the symbol * denotes the <u>multiplication</u> operation.**

Question 1 : C Arrays (8 marks)

Write a **full C program** having the,

i) user defined function `transformArray` that takes an integer array `arr` and its number of elements `n` as parameters. The function inspects the array **sequentially** to perform the following transformation on the array:

- If the value of an array element is **even**, divide it by 2.
- If the value of an array element is **odd**, replace it with the value of the next element in the array **if this next element exists**, otherwise set the odd value of the array element to 0.

[5 marks]

ii) `main()` function that initially defines an integer array `Data`. Next, it calls the function `transformArray` with `Data` and its number of elements as arguments. Finally, it prints the values of the array `Data`.

[3 marks]

Reminder: Don't forget the program headers, function prototypes, and any defined symbolic constants.

Question 2: Recursive function, Dynamic memory allocation, Pointers, String and Union (9.5 marks)

a) Write a C function code to return the value of the function f(x), where x is an integer:

$$f(x)=\begin{cases} x\ ,\ when\ x \leq 1 \\ f(x-1)+2*f(x-2)\ ,\ when\ x > 1 \end{cases}$$

[2.5 marks]

b) Write a C code fragment to,

i) request dynamically a memory segment to hold SIZE elements of integer values, where SIZE is a defined symbolic constant. Check if the request was successful and add subsequently relevant **printf** statements.

ii) set the values of the memory segment sequentially to 1, 3, 5, 7,…(arithmetic integer sequence with the general term a(i)=1+2*i with i ≥0) **using pointers, and not array indexing.**

[4 marks]

c) Consider the program of Figure Q2.

i) Give the printed value from the execution of statement of line 10. Explain your answer.

ii) Add a C statement in line 12 of the program to copy "BMW" to the `brand` member of the car variable.

iii) With line 12 of the program updated as required in question (ii), give the printed values from statements of Lines 14 and 15. Explain your answer.

```
1:  #include <stdio.h>
2:  #include<string.h>
3:  typedef union
4:  {
5:     char brand[12];
6:     float price;
7:  } uCar_t;
8:   int main()
9:  {
10:  printf("size of union type = %ld bytes", sizeof(uCar_t));
11 : uCar_t car;
12 :
13:  car.price=71000.0;
14:  printf("\n The car's brand is= %s",car.brand);
15:  printf("\n The price of the car is= %.2f",car.price);
 }
```

Figure Q2

[3 marks]

Question 3 : Struct and Pointers (9 marks)

Complete the code of Figure **Q3** as follows:

i) Write a **global** C struct datatype `Student_t` to represent a student record to hold a student's:

a. name as a C-string of 20 valid characters

b. age in years

c. GPA (grade point average)

[1.5 marks]

ii) The function void `update_gpa` has the parameter `st:` a **pointer** variable to `Student_t,` and . the `float` parameter `increment`. Inside the function, increment the GPA member of `st` with the value of the parameter `increment.`

[2 marks]

iii) Declare in the `main()` a `Student_t` variable `student`; keep prompting the user to input a valid student record (name, age, GPA) until (s)he does so; assign the valid input data to `student.` For simplicity assume that the values of the user's input, if of valid datatypes, are within the expected positive range. The input data, however, might be of wrong datatype; you can use the function `skip_line()` given in the appendix to deal with wrong data type input.

From the `main()` function, call `update_gpa` with the parameters values `student` and `0.20.`

The code for store_record_file() does not have to be completed.

[5.5 marks]

```
#include <stdio.h>

int main ()
{
....
 // store_record_file()
  return 0;
}

void upd_gpa(…){
…
}
```

Figure Q3.

Question 4: Binary Data File Processing (13.5 marks)

A C application stores in the binary formatted file `sensor.bin` a sensor readings of `float` values.

- Write a C **code fragment** to access `sensor.bin` to find the **maximum** value of its content. The file should be read block-by-block. The block size should be a defined symbolic constant of 128. **The number of values in the file is not necessarily a multiple of 128.** Include C statements to verify that,
  - the access request to the file was successful.
  - the end of file was reached.
  - No hardware failure occurred during the reading of the file.

Question 5: C++ Basic Programming (5 marks)

i) Write a C++ program consisting of the `main()` function **followed by** a C++ **template function** `void cube_bias`. The template C++ function has **a C++ <u>reference variable</u>** parameter `p1` and a **C++ <u>default parameter</u>** `b1` of value 10. All the parameters are of the same data type **T.** The `cube_bias` function updates `p1` with the sum of `b1` and the cube of `p1`, i.e. $(p1)^3 + b1$.

In the `main()` function, define a `float` variable `m`= 5.00 and call the function `cube_bias` with the parameter `m` and the bias value 20. Add a C++ statement to print the value of `m` after the function call.

[4 marks]

ii) What will be the resulting value of `m` after the call to the `void` function `cube_bias`. Explain your answer.

[1 mark]