TABLE OF SPECIFICATION
COMPUTER PROGRAMMING II MIDTERM EXAMINATION

2nd Semester, A.Y. 2024-2025

BSIT - 1B, 1C, 1D

		NUMBER OF ITEMS IN COGNITIVE LEVELS													ITEM PLACEMENT					
TOPICS	TIME ALLOTMENT (HOUR)	REMEMBERING (R)		UNDERSTANDING (U)		APPLYING (A)		ANALYZING (A)		EVALUATING (E)		CREATING (C)		TOTAL	R	U	A	An	E	C
		10%		20%		20%		20%		20%		10%								
ARRAY	15	1.80	2	3.60	4	3.60	4	3.60	4	3.60	4	1.80	2	36	1 -2	7-10	19 - 22	31 - 34	43 - 46	55 - 56
One Dimensional Array																				
Two - Dimensional Array																				
Multidimensional Array																				
Functions	15	1.80	2	3.60	4	3.60	4	3.60	4	3.60	3	1.80	2	35	3-4	11 - 14	23 - 26	35- 38	47 - 49	57 - 58
Built In Functions	3	0.36	0	0.72	0	0.72	0	0.72	0	0.72	1	0.36	0	5					50	
Structures	17	2.04	2	4.08	4	4.08	4	4.08	4	4.08	4	2.04	2	38	5-6	15 - 18	27 - 30	39 - 42	51 - 54	59 - 60
TOTAL	50	6.00	**6**	12.00	**12**	12.00	**12**	12.00	**12**	12.00	**12**	6.00	**6**	60						

Prepared:

MARY GRACE L. PEROCHO
Course Facilitator

Approved

CERILO B. RUBEN, MIT
Program Chairman, BSIT

Date: March 10, 2024

Republic of the Philippines
SULTAN KUDARAT STATE UNIVERSITY
COLLEGE OF COMPUTER STUDIES
Isulan Campus
Isulan, Sultan Kudarat

Name: ________________________________ Year / Section: _______________

CC113 – COMPUTER PROGRAMMING II

MIDTERM EXAMINATION

March 19 – 21, 2025

1. What is the term for the position of an element within an array?
   a. Value b. Address c. Index d. Size
2. What is the definition of an array?
   a. A single variable that stores one value.
   b. A collection of different data types.
   c. A group of similar kinds of elements stored in contiguous memory.
   d. A function that manipulates data.
3. What is the purpose of a function in C++?
   a. To declare variables
   b. To group statements that perform a specific task
   c. c) To define data types
   d. d) To manage memory allocation
4. What does a function declaration tell the compiler?
   a. The function's body
   b. The function's name, return type, and parameters
   c. The function's local variables
   d. The function's memory address
5. What keyword is used to define a structure in C++?
   a. class b) union c) struct d) typedef
6. Which of the following best describes a C++ structure?
   a. A collection of variables of the same data type.
   b. A collection of variables of different data types under a single name.
   c. A function that returns multiple values.
   d. A loop that iterates through a set of instructions.
7. In a two-dimensional array, what do the two indexes represent?
   a. Length and width c. Row and column
   b. Height and depth d. Value and address
8. How are arrays typically passed to functions in C++?
   a. By reference, passing the memory address of the first element.
   b. By value, copying all elements.
   c. By creating a new array within the function.
   d. By passing each element individually.
9. What does it mean to "traverse" an array?
   a. To delete all elements of the array. c. To change the data type of the array.
   b. To access each array element in sequence. d. To sort the array in reverse order.
10. Why is it often necessary to use loops when working with arrays?
    a. To change the data type of the array.
    b. To create a new array.
    c. To delete the array from memory.
    d. To access or manipulate multiple elements of the array efficiently.
11. What is the difference between a function declaration and a function definition?
    a. Declaration provides the body, definition provides the name.
    b. Declaration and definition are the same thing.
    c. Declaration provides the name and signature, definition provides the body.
    d. Declaration is for built-in functions, definition is for user-defined functions.
12. In "call by value," what happens to the original value of a variable passed to a function?
    a. It is copied, and changes within the function do not affect the original.
    b. It is modified within the function.

c. It is deleted.
d. It becomes a global variable.

13. What is the purpose of function overloading?
    a. To create functions with different names but the same parameters.
    b. To create functions that call themselves.
    c. To create multiple functions with the same name but different parameter lists.
    d. To create functions with default parameters.
14. Why is it important to provide a function declaration before the function is called?
    a. To allocate memory for the function.
    b. To execute the function's body.
    c. To define the function's local variables.
    d. To inform the compiler about the function's interface.
15. If you have a structure containing an array as a member, how are the individual elements of that array accessed within a structure variable?
    a. Using the structure variable name followed by the array name and index in square brackets.
    b. Using the array name followed by the structure variable name and index in square brackets.
    c. Using the structure variable name and index in square brackets, followed by the array name.
    d. Using the array name and index in square brackets, without the structure variable name.
16. What is the primary purpose of using a structure in C++ when dealing with related data?
    a. To perform mathematical calculations on the data.
    b. To group related data items of different data types into a single unit.
    c. To sort data in ascending or descending order.
    d. To access data directly from a database.
17. Which of the following statements accurately describes the fundamental difference between a C++ structure and a C++ array?
    a. Structures can only store integer data, while arrays can store any data type.
    b. Structures store elements of the same data type, while arrays store elements of different data types.
    c. Structures group variables of different data types under one name, while arrays store elements of the same data type.
    d. Structures are used for dynamic memory allocation, while arrays are used for static memory allocation.
18. What is the purpose of the dot operator (.) when working with C++ structures?
    a. To declare a structure variable.
    b. To define a structure.
    c. To access individual members of a structure variable.
    d. To initialize all members of a structure variable at once.
19. Given the declaration int numbers[5] = {10, 20, 30, 40, 50};, how would you access the third element (30)?
    a. numbers[3] b. numbers[2] c. numbers[1] d. numbers[4]
20. To sum the elements of a one-dimensional array values, what programming construct is most commonly used?
    a. if-else statement c. for loop
    b. switch statement d. while loop
21. Given the array int values[] = {5, 10, 15, 20};, what is the result of values[1] + values[3]?
    a. 20 b. 25 c. 30 d. 35
22. If you have a two-dimensional array matrix[3][4], how many elements does it contain?
    a. 7 b. 12 c.9 d. 16
23. Given the function declaration int calculate(int a, int b = 10);, what will be the result of calculate(5);?
    a. 7 b. 10 c.15 d. Error
24. You need to create a function that can add two numbers, whether they are integers or doubles. Which C++ feature would be most suitable?
    a. Default parameters c. Call by pointer
    b. Function overloading d. Recursion
25. Write the function header of a function named "multiply" that takes two double parameters and returns a double.
    a. void multiply(double a, double b); c. multiply(double a, double b);
    b. double multiply(double a, double b); d. double multiply(a, b);

26. Given the function void modify(int *ptr) { *ptr = *ptr * 2; }, if int x = 5;, what will be the value of x after calling modify(&x);?

a. 5 b. 10 c. 2.5 d. An Error

27. You need to create an array of 5 Book structures and initialize the first book with the title "C++ Basics", 300 pages, and a price of 29.99. Which of the following code snippets correctly performs this initialization?

```
struct Book {
   string title;
  int pageCount;
 float price;
};
```

a. Book books[5];
books[0] = {"C++ Basics", 300, 29.99};
b. Book books[5] = {"C++ Basics", 300, 29.99};
c. Book books[5];
books = {"C++ Basics", 300, 29.99};
d. Book books[5] = {{"C++ Basics"}, {300}, {29.99}};

28. Write a function printDate that takes a Date structure as input and prints the date in the format "day/month/year". Which of the following function implementations is correct?

```
struct Date {
   int day;
   int month;
   int year;
};
```

a. void printDate(Date d) {
cout << d.day << "-" << d.month << "-" << d.year << endl;}
b. void printDate(Date d) {
cout << d.day << "/" << d.month << "/" << d.year << endl;}
c. void printDate(Date *d) {
cout << *d.day << "/" << *d.month << "/" << *d.year << endl;}
d. void printDate(Date d) {
printf("%d/%d/%d", d.day, d.month, d.year);}

29. A program requires storing information about books, including title (string), author (string), and publication year (integer). Which of the following C++ structure declarations would be most appropriate?

a. struct Book { int title; int author; int year; };
b. struct Book { string title; string author; int year; };
c. struct Book (string title, string author, int year);
d. struct Book [string title, string author, int year];

30. *struct Point {*

```
struct Point {
        int x;
        int y;
};
```

And the structure variable *Point p1*;, which of the following code correctly sets the x coordinate of p1 to 10?

a. p1(x) = 10; b. p1->x = 10; c. p1.x = 10; d. p1[x] = 10;

31. Which of the following best describes the value stored in the variable value and why?

int data[5] = {10, 20, 30, 40, 50}; int *ptr = data + 2; int value = *(ptr - 1);

a. 30, because ptr points to the third element, and subtracting 1 accesses the second element.
b. 20, because ptr points to the third element, and subtracting 1 accesses the second element.
c. 40, because ptr points to the third element, and subtracting 1 accesses the fourth element.
d. 10, because ptr points to the first element, and subtracting 1 accesses the last element.

32. A program needs to store daily temperature readings for a month (30 days) and then calculate the average temperature. Which array type and access method would be most efficient and why?

a. A two-dimensional array, accessed using nested loops, for easy row and column management.
b. A one-dimensional array, accessed using a single loop, for direct sequential access and efficient calculation.
c. A multi-dimensional array with three dimensions, to allow for more complex analysis.
d. A linked list, because linked list are always more efficient than arrays.

33. Given the following 2D array and code:

int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}; int sum = 0; for (int i = 0; i < 3; i++) { sum += matrix[i][i]; }

What does the code calculate, and what is the final value of sum?

a. It calculates the sum of all elements in the array; sum = 45.
b. It calculates the sum of the diagonal elements; sum = 15.
c. It calculates the sum of the first row; sum = 6.

   d. It calculates the sum of the middle column; sum = 15.

34. In a seat reservation system using a 2D array seats[rows][cols], where true represents an available seat and false represents a reserved seat, what logical operation would you use to check if a specific seat (row r, column c) is available before reserving it, and why?
   a. seats[r][c] = false, because it assigns the seat as reserved.
   b. seats[r][c] == true, because it checks if the seat is available.
   c. seats[r][c] = true, because it assigns the seat as available.
   d. seats[r][c] != false, because it checks if the seat is reserved.

35. Consider the given code at the right. What is the primary purpose of the process function?
   a. To calculate the sum of two numbers.
   b. To calculate the absolute difference between two numbers.
   c. To determine the larger of two numbers.
   d. To swap the values of two numbers.

```
int process(int a, int b) {
   if (a > b) {
      return a - b;
   } else {
      return b - a;
   }
}
```

36. Analyze the following overloaded functions:

```
int calculate(int x, int y);
double calculate(double x, double y);
```

What is the key difference that allows the compiler to distinguish between these two functions?
   a. The function names.
   b. The return types.
   c. The parameter data types.
   d. The order of the parameters.

37. Consider the following code containing a swap function:

```
void swap(int *x, int *y) {
   int temp = *x;
   *x = *y;
   *y = temp;
}
```

What mechanism does this function utilize to modify the original variables passed to it?
   a. Call by value.
   b. Call by reference.
   c. Call by pointer.
   d. Function overloading.

38. When analyzing a program that uses both recursive and iterative functions for similar tasks, what factor would you consider when comparing their performance?
   a. The amount of code.
   b. Stack space usage and execution time.
   c. The number of parameters.
   d. The function's return type.

39. Consider a scenario where you have a nested structure to represent a student's record, including their personal information and address:

```
struct Address {
        string street;
        string city;
};
struct Student {
        string name;
         int studentID;
        Address homeAddress;
};
```

If you have a Student variable named std1, how would you access the city member of the homeAddress field?
   a. std1.city
   b. std1.homeAddress.city
   c. std1->homeAddress->city
   d. std1.city.homeAddress

40. These two C++ structure designs for storing a person's name and age:

```
// Design 1
struct Person1 {
    string fullName;
    int age;

};
```

```
// Design 2
struct Person2 {
    string firstName;
    string lastName;
    int age;
};
```

Evaluate which design is better for a system that frequently needs to search for people by their last name.

a. Design 1 is better because it uses less memory.
b. Design 2 is better because it allows direct access to the last name for searching.
c. Design 1 is better because it simplifies printing the full name.
d. Both designs are equally effective for searching by last name.

41. These two options of structure designed to store book information. C++

```
// Option 1
  struct Book1 {
        string title;
        string author;
        int publicationYear;
  };
```

```
// Option 2
  struct Book2 {
        string title;
        struct Author {
  string firstName;
        string lastName;
  } author;
  int publicationYear;
  };
```

Evaluate which option is better if you need to frequently access and display the author's full name.

a. Option 1 is better because it simplifies the code needed to display the full name.
b. Option 2 is better because it logically groups the author's first and last names.
c. Option 1 is better because it uses less memory.
d. Both options are equally effective for displaying the author's full name.

42. These two options of structure designed to store product information.

// Structure 1

```
struct Product1 {
   int productID;
   string productName;
   double price;
};
```

// Structure 2

```
struct Product2 {
   int productID;
   string productDetails[2]; // productDetails[0] = productName, productDetails[1] = price
};
```

Evaluate which structure is better for readability and maintainability, especially if the number of product details might increase in the future.

a. Structure 1 is better because it uses named members for each detail, making the code more readable.
b. Structure 2 is better because it uses less memory.
c. Structure 2 is better because it simplifies adding new product details.
d. Both structures are equally effective for readability and maintainability.

43. In what scenarios would using a multidimensional array be more appropriate than using multiple one-dimensional arrays?

a. dealing with a simple list of items.
b. When storing data that naturally fits into a table or matrix format.
c. When only needing to store a small number of elements.
d. When needing to perform linear searches frequently.

44. A developer is considering using a one-dimensional array versus a two-dimensional array to store student grades for multiple subjects. Which of the following evaluations best justifies the choice of a two-dimensional array based on the information provided?

a. A one-dimensional array is always more efficient due to its simpler structure and single index.

b. A two-dimensional array is suitable because it effectively represents students as rows and subjects as columns, aligning with the "rows and columns that form a matrix" concept mentioned.
c. The choice is arbitrary, as both array types can store the data equally well without significant difference.
d. One dimensional array is better because it uses less memory.

45. A programmer is tasked with storing survey results where participants rate a product on a scale of 1 to 5. They are considering using an array to store the frequency of each rating. Given the information about array characteristics, which of the following evaluations best justifies using an array for this task?
    a. Arrays are unsuitable for counting frequencies because they are designed for storing single values.
    b. Using an array where the index represents the rating (1 to 5) and the value at that index represents the frequency is an efficient way to summarize the survey results, aligning with the concept of "Using Elements of an Array as Counters" and the need to store and access "a collection of data... of the same type."
    c. Arrays can only store the raw survey results and cannot be used to calculate frequencies.
    d. Linked lists are better suited for counting frequencies than arrays.
46. Assess the efficiency of using a linear search on a large, sorted array. What are the limitations?
    a. Highly efficient because it can quickly find the element.
    b. Efficient for small arrays but inefficient for large arrays due to its time complexity.
    c. Efficient for sorted arrays but inefficient for unsorted arrays.
    d. Always efficient regardless of array size or order.
47. In what scenario would the use of default parameters in a function be considered most beneficial?
    a. When the function always requires all parameters.
    b. When the function has no parameters.
    c. When some parameters have commonly used values.
    d. When the function is recursive.
48. Evaluate the use of function overloading in a large software project where multiple teams contribute to the code.
    a. It leads to code duplication and confusion.
    b. It should be avoided due to the potential for naming conflicts.
    c. Its effectiveness depends on the compiler's optimization capabilities.
    d. It enhances code reusability and clarity if used consistently.
49. Evaluate the effectiveness of using "call by reference" versus "call by value" when dealing with large data structures as function parameters.
    a. "Call by value" is more efficient due to data copying.
    b. "Call by reference" is more efficient as it avoids unnecessary copying.
    c. Both are equally efficient.
    d. The efficiency depends on the function's return type.
50. In terms of efficiency, which of the following is generally true when using built-in C++ string manipulation functions compared to writing custom loops?
    a. Custom loops are always more efficient due to direct control.
    b. Built-in functions are often optimized and therefore more efficient.
    c. Efficiency depends entirely on the specific compiler.
    d. There is no significant difference in efficiency.
51. The provided text states that a structure is similar to a class. Given that both can hold collections of different data types, under what circumstances would using a structure be more appropriate than a class in a C++ program?
    a. When encapsulation and data hiding are critical program requirements.
    b. When the primary need is to group related data without complex member functions.
    c. When inheritance and polymorphism are essential for the program design.
    d. When creating a complex, object-oriented hierarchy.
52. You need to access a specific student's grade from an array of student structures. Which method is the most efficient and readable, and why?
    a. Accessing the grade using a loop that iterates through the entire array.

b. Accessing the grade using the dot operator with the array index (e.g., students[index].grade).
c. Accessing the grade by creating a temporary copy of the structure.
d. Accessing the grade by converting the structure to a string and then parsing the grade.

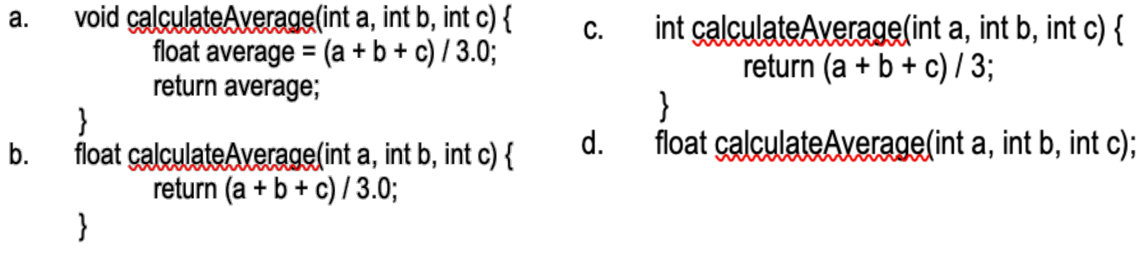
53. You are designing a program to store employee data. You have the option to use either separate variables for each employee's information (name, ID, salary) or a structure. Which of the following best justifies the use of a structure in this scenario?
a. Using a structure will reduce the amount of memory used.
b. Using a structure will make the program run faster.
c. Using a structure will improve the organization and readability of the code by grouping related data.
d. Using a structure is necessary to store data in a database.

54. When designing a structure to store a person's age, which data type would be the most appropriate, and why?
a. double, because it can store decimal values.
b. string, because it can store any character sequence.
c. int, because age is a whole number within a limited range.
d. long double, because it provides the highest precision.

55. Design a one-dimensional array to store the number of days in each month of a non-leap year. Initialize it with the appropriate values.
a. int days[12] = {30, 31, 30, 31, 30, 31, 30, 31, 30, 31, 30, 31};
b. int days[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
c. int days[12] = {31, 30, 31, 30, 31, 30, 31, 30, 31, 30, 31, 30};
d. int days[12] = {28, 30, 31, 30, 31, 30, 31, 30, 31, 30, 31, 30};

56. A developer needs to store a collection of student names. They are debating between using a single one-dimensional array of strings or creating separate string variables for each student. Which of the following evaluations best justifies the choice of using a one-dimensional array?
a. Using separate string variables is more efficient because it avoids the overhead of array indexing.
b. Using a one-dimensional array is justified because it allows for efficient storage and traversal of a collection of similar data (student names), aligning with the need to "store a large amount of data of similar type" and the concept of "traversing an array."
c. Using separate string variables offers better memory management because it allocates memory only when needed.
d. There is no significant difference between using an array or separate variables for storing student names.

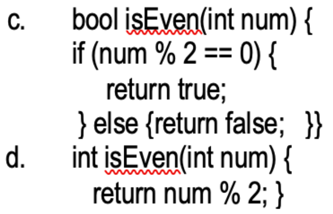
57. Design a C++ function named calculateAverage that takes three integer parameters and returns their average as a floating-point number. Which of the following function prototypes and definitions BEST represents this function?



a.
```
void calculateAverage(int a, int b, int c) {
        float average = (a + b + c) / 3.0;
        return average;
}
```
b.
```
float calculateAverage(int a, int b, int c) {
        return (a + b + c) / 3.0;
}
```
c.
```
int calculateAverage(int a, int b, int c) {
        return (a + b + c) / 3;
}
```
d.
```
float calculateAverage(int a, int b, int c);
```

58. Create a C++ function named isEven that takes an integer parameter and returns a boolean value indicating whether the number is even. Which of the following function implementations BEST fulfills this requirement?

a.
```
void isEven(int num) {
            if (num % 2 == 0) {
             cout << "Even";   } }
```
b.
```
bool isEven(int num) {
             return num / 2;}
```



c.
```
bool isEven(int num) {
if (num % 2 == 0) {
    return true;
 } else {return false;   }}
```
d.
```
int isEven(int num) {
  return num % 2; }
```

59. Design a structure named Rectangle that stores the width and height of a rectangle as floating-point numbers. Then, write a function that takes a Rectangle structure as input and calculates and returns the area of the rectangle. Which of the following code snippets correctly implements this?

a.
```
struct Rectangle {
        float width;
        float height;
};
float calculateArea(Rectangle rect) {
        return rect.width * rect.height;
}
```

b.
```
struct Rectangle {
        int width;
        int height;
};
int calculateArea(Rectangle rect) {
        return rect.width * rect.height;
}
```

c.
```
struct Rectangle {
        float width, height;
};
void calculateArea(Rectangle rect) {
        cout << rect.width * rect.height;
}
```

d.
```
struct Rectangle {
        float width;
        float height;
};

void calculateArea(float width, float height) {
        cout << width * height;
}
```

60. A software developer is tasked with creating a system to manage employee data. Each employee has a unique employee ID (integer), a name (string), a salary (double), and a date of hire (which should store day, month, and year as integers).Design a C++ structure that effectively represents this employee data. Which of the following code snippets BEST fulfills this requirement?

a.
```
struct Employee {
    int employeeID;
    string name;
    double salary;
    int hireDay;
    int hireMonth;
    int hireYear;
};
```

b.
```
struct Date {
    int day;
    int month;
    int year;
};

struct Employee {
    int employeeID;
    string name;
    double salary;
    Date hireDate;
};
```

c.
```
struct Employee {
    int employeeID;
    string name;
    double salary;
    string hireDate;
};
```

d.
```
struct Employee {
    int employeeID;
    string name;
    double salary;
    int hireDate[3];
};
```

Prepared by:

MARY GRACE L. PEROCHO