



University of Fort Hare  
*Together in Excellence*

**DEGREE JUNE EXAMINATION 2023**

SUBJECT	:	ADVANCED PROGRAMMING IN C++
SUBJECT CODE	:	CSC211
QUALIFICATION	:	BACHELOR OF SCIENCE
EXAMINERS	:	MR. S. NGWENYA
MODERATOR	:	MR. P. NOMNGA
DURATION	:	3 hrs
MARKS	:	100
NUMBER OF PAGES	:	7(Including cover page)

**INSTRUCTIONS:**

- Read the questions carefully and make sure that all questions are clear to you.
- This is a closed book test
- No student is allowed to borrow a pen, pencil, eraser, etc. from another student.
- Questions must be referred to the invigilator or subject lecturer only.
- All questions should be answered on the paper provided.
- Ensure that your last name and student number appear at the top of each page used.
- NB: Cell phones must be switched off.

**Question One (20 Marks)**

1. The components of a struct are called the \_\_\_\_ of the struct.
  - a. variables
  - b. identifiers
  - c. elements
  - d. members
2. Which of the following struct definitions is correct in C++?
  - a. 

```
struct studentType
{
    int ID;
};
```
  - b. 

```
struct studentType
{
    string name;
    int ID;
    double gpa;
}
```
  - c. 

```
int struct studentType
{
    ID;
}
```
  - d. 

```
struct studentType
{
    int ID = 1;
};
```
3. Consider the following struct definition:

```
struct rectangleData
{
    double length;
    double width;
    double area;
    double perimeter;
};
```

Which of the following variable declarations is correct?
  - a. `rectangle rectangleData;`
  - b. `struct rectangleData();`
  - c. `rectangleData myRectangle;`
  - d. `rectangleData rectangle = new rectangleData();`
4. Typically, in a program, a struct is defined \_\_\_\_ in the program.
  - a. in the main function
  - b. before the definitions of all the functions
  - c. after the definitions of all the functions
  - d. in any function
5. An array name and index are separated using \_\_\_\_.
  - a. curly brackets
  - b. square brackets

c. a dot d. a comma

6. The syntax for accessing a struct member is structVariableName\_\_\_\_\_.

- a. .memberName b. \*memberName  
c. [memberName]d. \$memberName

7. Consider the following statements:

```
struct rectangleData  
{  
    double length;  
    double width;  
    double area;  
    double perimeter;  
};
```

```
rectangleData bigRect;
```

Which of the following statements correctly initializes the component length of bigRect?

- a. bigRect = {10}; b. bigRect.length = 10;  
c. length[0]= 10; d. bigRect[0]= 10

8. In C++, the \_\_\_\_\_ symbol is an operator, called the member access operator.

- a. :(colon) b. .(dot)  
c. ,(comma) d. \$ (dollar sign)

9. Consider the following statements:

```
struct rectangleData  
{ double length;  
    double width;  
    double area;  
    double perimeter;  
};
```

```
rectangleData bigRect;
```

Which of the following statements is valid in C++?

- a. cin >> bigRect;  
b. cin >> bigRect.length;  
c. perimeter = 2 \* (length + width);  
d. area = length \* width;

10. Consider the following statements:

```
struct personallInfo  
{  
    string name;  
    int age;  
    double height;  
    double weight;  
};
```

```
struct commonInfo  
{  
    string name;  
    int age;
```

```
};
```

```
personInfo person1, person2;
```

```
commonInfo person3, person4;
```

Which of the following statements is valid in C++?

a. person1 = person3;      b. person2 = person1;

c. person2 = person3;      d. person2 = person4;

11. A struct is a(n) \_\_\_\_\_, not a declaration.

12. The following statement defines a struct houseType with a total of \_\_\_\_\_ member(s).

```
struct houseType
{
    string style;
    int numOfBedrooms;
    int numOfBathrooms;
    int numOfCarsGarage;
    int yearBuilt;
};
```

13. Memory is allocated for struct variables only when you \_\_\_\_\_ them.

14. Arrays are passed by \_\_\_\_\_ only.

15. Consider the following struct definition:

```
const int ARRAY_SIZE = 1000;
struct listType
{
    int listElem[ARRAY_SIZE];
    int listLength;
};
```

The statement that declares intList to be a struct variable of type listType is \_\_\_\_\_.

16. If a variable is passed by \_\_\_\_\_, then when the formal parameter changes, the actual parameter also changes.

17. When you attach & after the dataType in the formal parameter list of a function, the variable following that dataType becomes a(n) \_\_\_\_\_ parameter.

18. \_\_\_\_\_ parameters are useful in three situations:

- When the value of the actual parameter needs to be changed
- When you want to return more than one value from a function
- When passing the address would save memory space and time relative to copying a large amount of data

19. Stream variables (for example, ifstream and ofstream) should be passed by \_\_\_\_\_ to a function.

20. If a function needs to return more than one value, as a rule of good programming style, you should change it to a(n) \_\_\_\_\_ function and use the appropriate reference parameters to return the values.

### Question Two (20 Marks)

1. Suppose `x` and `y` are `int` variables and `ch` is a `char` variable. Consider the following input: [10]  
5 28 36  
What value (if any) is assigned to `x`, `y`, and `ch` after each of the following statements executes? (Use the same input for each statement.)
  - a. `cin >> x >> y >> ch;`
  - b. `cin >> ch >> x >> y;`
  - c. `cin >> x >> ch >> y;`
  - d. `cin >> x >> y;`  
`cin.get(ch);`
  
2. Write C++ statements to do the following: [10]
  - a. Open the file `travel.dat` using the variable `outfile`.
  - b. Write the statement to format your output to two decimal places in fixed form.
  - c. Write the values of the variables `day`, `distance`, and `speed` in the file `travel.dat`.
  - d. Calculate and write the `travelTime` in the file `travel.dat`.
  - e. Which header files are needed to process the information in (a) to (d)?

### Question Three (20 Marks)

1. Suppose that `x`, `y`, `z`, and `w` are `int` variables, and `x = 3`, `y = 4`, `z = 7`, and `w = 1`. What is the output of the following statements? [2]
  - a. `cout << "x == y: " << (x == y) << endl;`
2. `int count = 1;`  
`do`  
`cout << count *(count - 2) << " ";`  
`while (count++ <= 5);`  
`cout << endl;` [2]
3. `for (i = 1; i <= 1; i++)`  
`cout << "*";`  
`cout << endl;` [2]
4. Consider the following function: [2]

```
int mystery(int x, double y, char ch)
{
    int u;
    if ('A' <= ch && ch <= 'R')
        return(2 * x + static_cast<int>(y));
    else
        return(static_cast<int>(2 * y) - x);
}
```

What is the output of the following C++ statements?

- a. `cout << mystery(5, 4.3, 'B') << endl;` [2]
5. Explain the difference between an actual and a formal parameter. [2]
6. Write C++ statements that do the following: [10]
- Define an enum type, `bookType`, with the values `MATH`, `CSC`, `ENGLISH`, `HISTORY`, `PHYSICS`, and `PHILOSOPHY`.
  - Declare a variable `book` of type `bookType`.
  - Assign `MATH` to the variable `book`.
  - Advance `book` to the next value in the list.
  - Output the value of the variable `book`.

#### Question Four (20 Marks)

1. Write C++ statements that do the following: [10]
- Declare an array `alpha` of 10 rows and 20 columns of type `int`.
  - Initialize the array `alpha` to 0.
  - Store 1 in the first row and 2 in the remaining rows.
  - Store 5 in the first column, and make sure that the value in each subsequent column is twice the value in the previous column.
  - Print the array `alpha` one row per line.
- Possible Answers

2. Given the declaration:

```
char str1[15];
```

```
char str2[15] = "Good day";
```

mark the following statements as valid or invalid. If a statement is invalid, explain why.

a. `str1 = str2;`

b. `if (str1 == str2)`

```
cout << " Both strings are of the same length." << endl;
```

c. `if (strlen(str1) >= strlen(str2))`

```
str1 = str2;
```

d. `if (strcmp(str1, str2) < 0)`

```
cout << "str1 is less that str2." << endl;
```

[10]

**Question Five (20 Marks)**

1. Write a program that uses the function `isNumPalindrome` given in Example 6-5 (Palindrome Number). Test your program on the following numbers:  
10, 34, 22, 333, 678, 67876, 44444, and 123454321. [5]
2. Write a value-returning function, `isVowel`, that returns the value `true` if a given character is a vowel and otherwise returns `false`. [5]
3. Write a program that prompts the user to input a sequence of characters and outputs the number of vowels. (Use the function `isVowel` written in Programming Exercise 2.) [5]
4. Write a C++ function, `smallestIndex`, that takes as parameters an `int` array and its size and returns the index of the first occurrence of the smallest element in the array. Also, write a program to test your function. [5]