

OLABISI ONABANJO UNIVERSITY, AGO-IWOYE
FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICAL SCIENCES
B.Sc. DEGREE EXAMINATIONS (300 LEVEL)
HARMATTAN SEMESTER 2017/2018

CMP 301: OBJECT-ORIENTED PROGRAMMING

INSTRUCTIONS: Answer ANY four questions
All questions carry equal marks

Time Allowed: 2hrs.

Question 1

- (a) Rewrite the following program using C++ style I/O

```
#include<stdio.h>
int main(void) {
    int a, b, d, min;
    printf("Enter two numbers:");
    scanf("%d %d", &a, &b);
    min = a > b ? b : a;
    for(d = 2; d < min; d++)
        if(((a%d==0) && ((b%d==0)))break;
    if(d==min) {
        printf("No common denominator\n");
        return 0;
    }
    printf("The lower common denominator
is %d\n");
    return 0;
}
```

- (a) Give the format for declaring a class structure in C++.
- (c) For each of the following, state whether a *void* or *nonvoid* method is the most appropriate
- (i) Computing a square root
 - (ii) Printing a message a number of times
 - (iii) Computing a sales commission, given the sales amount and the commission rate

- (iv) Testing whether a number is *even* and returning **true** if it is
- (v) Printing a calendar for a month

Question 2

- (a) Briefly discuss the concept of *encapsulation* as one of the cornerstones of object-oriented programming.
- (b) Use a full-fledge sample code to illustrate the implementation of the concept in (a) above.

- (c) What is the problem with the following code fragment?

```
#include<iostream>
using namespace std;
main() {
    myclass ob1, ob2;
    ob1.a = 100;
    ob2.a = 90;
    cout << ob1.get_a() << "\n";
    cout << ob2.get_a() << "\n";
} return 0;
```

Question 3

- (a) Explain *function prototyping*. What is its importance in C++ programming?
- (b) Given three integer variables *a*, *b*, *c* write a C++ program using a function **sum** to add two integer values and then return the result to *c*.

Turn Over

(c) Will the following program compile?

Justify

```
#include<iostream>
using namespace std;
main() {
    foo;
    return 0;
}
void foo() {
    cout << "This is a function\n";
}
```

Question 4

- (a) (i) Briefly discuss **function overloading** as a form of *polymorphism* in object-oriented programming.
- (ii) Give the necessary and sufficient condition for overloading a function.
- (b) Develop an overloaded function **Date()** to accept either a string input or three integer variables.
- (c) The C++ standard library contains the following three functions:

```
double atof(const char *s)
int atoi(const char *s)
long atoi(const char *s)
```

Is it possible to overload these functions?

Explain.

Question 5

- (a) State the difference between a constructor function and other functions.
- (b) Create a class **box** whose constructor function is passed three *double* values *length*, *width* and *height*. Have the **box** class compute the *volume* of the box and store the result in a *double* variable.

Include a member function **vol** that displays the volume of the box.

- (c) (i) Define *class inheritance* in OOP.
- (ii) Differentiate between the two types of class inheritance in OOP with well-illustrated examples.

Question 6

- (a) (i) Briefly explain *operator overloading*.
- (ii) State the rules that apply to operator overloading.
- (b) Give the summary of operators that may be overloaded.