

**COS1512
RCO1512**

October/November 2012

INTRODUCTION TO PROGRAMMING 2

Duration 2 Hours

75 Marks

EXAMINERSFIRST
SECONDMISS CNA NOMBEWU
PROF ID SANDERS

MRS MA SCHOEMAN

Closed book examination.

This examination question paper remains the property of the University of South Africa and may not be removed from the examination venue.

This paper consists of 6 pages.

**This examination paper remains the property of the University of South Africa
and may not be removed from the examination room.**

Instructions / Instruksies:

- 1 Answer all questions
- 2 All rough work must be done in your answer book
- 3 The mark for each question is given in brackets next to the question
4. Please answer the questions in order If you want to do a question later, leave enough space

GOOD LUCK!

QUESTION 1**[5]**

- 1 1 Consider the following declaration (1)
- ```
char number[] = "thirteen",
```

What is the maximum string length that can be stored in number? Explain your answer

- 1 2 What would be the expected output from the following? (2)
- ```
char funny[15] = "Roley Poley",
int k = 0;
while(funny[k] != '\0')
{
    if (funny[k] == 'y')
        funny[k] = '*';
    k++;
}
cout<<funny,
```

- 1 3 Consider the following incomplete code fragment and answer the questions below (2)
- ```
1. vector<int> number(4);
2. number[0] = 3;
3. for (int k = 1, k < number.size(), k++)
4. {
5. number[k] = 2 + number[k - 1],
6. cout << number[k] << " ";
7. }
8
9. cout << number[number.size()-1],
```

1 3 1 Explain in one sentence what happens in line 1

1 3 2 For line 8, give the C++ instruction to add another element to number with the value of 11

1 3 3 What will the output of the code fragment be?

**QUESTION 2****[5]**

- 2 1 The following code fragment has some errors. Read the questions below and give the correct code for the lines in error:

```
1 int * p1,
2 int * p2;
3 int x = 5,
4 int y = 10;
5 *p1 = x;
6 *p2 = y,
7 cout << p1 << " " << p2 ;
8. p1 = p2,
9. cout << *p1 + *p2 + 10;
```

- 2 1.1 In lines 5 and 6 p1 and p2 are supposed to point to the variables x and y respectively. Give the correct code for lines 5 and 6 (1)

2 1.2 Line 7 is supposed to display the values of the variables to which p1 and p2 are pointing. Give the correct coding for line 7 (1)

2.1 3 What is displayed by line 9? (1)

2 2. Give an example of a static array definition (1)

2 3 Give an example of a dynamic variable (1)

### QUESTION 3 [35]

Define a class `Employee` that represents an employee working for a company on the JSE This class has three member variables

- `name`, a string that holds the name of the employee
- `nrYrsService`, an integer value that indicates the number of years this employee has been working for the company and
- `salary`, a float value that indicates the salary of the employee

In addition, the class should have the following member functions

- A **default constructor** that initializes `name` to an empty string `nrYrsService` and `salary` should each be initialized to 0
- An **overloaded constructor** to set `name`, `nrYrsService` and `salary` to specified values
- A **destructor** that does not perform any action
- Accessor functions for the member variables
- An **overloaded operator>** to compare two `Employee` objects The `operator>` is implemented as a **friend** function with the following prototype  

```
bool operator>(const Employee & e1, const Employee & e2)
```

This function returns true if `nrYrsService` for `e1` has a larger value than `nrYrsService` for `e2`, and false if not
- An overloaded **operator++** for class `Employee` that increments the value of the `nrYrsService` member variable by one Use the following prototype  

```
Employee operator++ (Employee E);
```
- An overloaded extraction **operator >>** (implemented as a **friend** function) so that it can be used to input values of type `Employee`
- An overloaded insertion **operator <<** (implemented as a **friend** function) that outputs all the member variables of an `Employee` object

You should attempt the solutions as follows

3 1 Create the header file `Employee.h` that contains the `Employee` class specification (8)

3 2 Create the implementation of the class `Employee` including all the **friend** functions (15)

3 3 Complete the application program (`main()`) below by citing the number and writing down the missing statement This program extracts all employees working for the company one by one from a file `Employees.dat` and updates the number of years' service for each using the `++` operator The program also determines the employee with the longest service All updated employees are output to a new file called `UpdatedEmployees.dat` Finally the program displays the name of the employee with the longest service on the console window Be sure to use appropriate member functions in the required statements

For example, if the input file has the following contents

**Input file (Employees.dat)**  
Peter 5 25000.50  
Sally 9 35000.75  
Hester 14 29530 48  
Joel 3 19000 35

The program should produce the following output

**Console output:**

Peter Bonus R 12506 25

Sally Bonus R 17510 38

Hester Bonus: R 14780 24

Joel Bonus R 9504 17

Hester has been with the company for the longest time  
Press any key to continue . .

**Output file (UpdatedEmployees.dat)**

Peter 6 25000.5

Sally 10 35000.8

Hester 15 29530.5

Joel 4 19000.3

**NB Note that number 3 and number 4 below require two (similar) statements each.**

(12)

```
#include <iostream>
#include <iomanip>
```

```

_____ 1 _____ //1.Include files needed
using namespace std,

int main()
{
 _____ 2 _____ //2.Declare input and output files

 _____ 3 _____ //3.Open the file Employees.dat and check
 // that the file exists

 _____ 3 _____ //3.Open the file UpdatedEmployees.dat and
 // check that the file exists

 _____ 4 _____ //4.Initialise object longestEmployee used to find
 // employee with longest service time

 _____ 4 _____ //4.Instantiate object aEmployee to extract an employee
 //from file Employees.dat

 while (_____ 5 _____) //5. Extract an employee from file Employees.dat
 {
 if (_____ 6 _____) //6. Compare aEmployee with longestEmployee to
 //determine the employee who has been the
 //longest with the company
 {
 longestEmployee = aEmployee; //update longestEmployee
 }

 _____ 7 _____; //7.Update number of years of service
 cout.setf(ios::showpoint);
 cout.setf(ios::fixed);
 cout << setprecision(2),

```

```

cout << _____ 8 _____ //8. Display employee name
cout << "\tBonus: R "
 << _____ 9 _____ //9. Calculate and display bonus
 << endl << endl,
 _____ 10 _____ //10. Output updated employee to file
 _____ //UpdatedEmployees.dat
}
cout << _____ 11 _____ << " has been with the company for the longest
time" << endl; //11. Display name of employee who has been longest with the
 _____ //company

_____ 12 _____ //12. Close files

return 0,
}

```

**QUESTION 4****[15]**

Consider the following class

```

class Module
{
public:
 Module(),
 Module(string dept, string mCode, double tFee),
 void displayFee(ostream &out) const,
 string get_department() const,
 string get_moduleCode() const,
 double get_tuitionFee() const,
 void set_department(string d);
 void set_moduleCode(string m);
 void set_tuitionFee(double t),
private:
 string department,
 string moduleCode,
 double tuitionFee,
},

```

- 4.1 Derive a class LabModule from class Module. This class has an additional member variable, labFee. Class LabModule also has member functions, get\_labFee() and set\_labFee() to return member variable labFee and to update member variable labFee respectively. The class LabModule should override function displayFee() in order to display the fee for the module, which will be the sum of the labFee and the tuitionFee. Provide only the interface of class LabModule in terms of a header file. The header file should contain compiler directives to prevent multiple definitions. Assume that the interface of class Module is contained in an interface file called Module.h (7)
- 4.2 Implement the overloaded constructor for the class LabModule by invoking the base class constructor (3)
- 4.3 Implement the member function displayFee for the class LabModule (4)
- 4.4 Is function displayFee for the class LabModule an example of overloading? Explain your answer. (1)

**QUESTION 5****[10]**

A set is an unordered collection of zero or more elements with no duplicates. The class interface below stores a set of integer elements in a vector.

```
class Set
{
public:
 Set(); //default constructor
 void add (const int &i); //add i to set
 void remove (const int &i); //remove i from set
 void display (ostream &out = cout) const; //list elements in Set
 Set intersection (const Set &s) const; //returns intersection with s
 Set union (const Set &s) const; //return union with s
 Set difference (const Set &s); //returns difference with s
private:
 vector<int> v;
}
```

- 5.1 Write a template version of the Set interface so that it can store a set of any type, e.g. a set of chars or a set consisting of a user defined class. Do not provide an implementation. Provide only the interface. (4)
- 5.2 Implement the add member function of the template class Set. (5)
- 5.3 Provide a declaration for a Set of chars. (1)

**QUESTION 6****[5]**

Function `addValues()` below accepts as a parameter an integer array `array1` and an integer `n` greater than 0 and calculates the sum of the first `n` elements of the array.

```
int addValues(int array1[], int n)
{
 int sum = 0;
 if (n < 1)
 {
 cout << "The value of n cannot be less than 0";
 exit(1);
 }
 else if (n == 1)
 return a[0];
 else
 for (k = 0, k < n, k++)
 sum = sum + array1[k];
 return (sum);
}
```

- 6.1 Rewrite function `addValues()` as a recursive function `recursiveAdd()` and use `return (a[n-1] + addValues(a, n-1))`, as the general case. Make sure that you include a base case. (4)
- 6.2 What is the purpose of the general case in a recursive function? (1)