

University of Swaziland
Department Of Computer Science
Final Examination 2005

Title of paper: *C under Unix*

Course number: *CS344*

Time Allowed: *Three (3) hours*

Instructions:

- *Each question is worth 20 marks*
- *Answer any five (5) questions from questions 1 to 7*

This paper may not be opened until permission has been granted by the invigilator

Question 1-20 marks

- (i) State whether each of the following is true or false. If false explain why? [10 Marks]
- (a) All variables must be declared before they are used.
 - (b) C++ considers the variables **number** and **Number** to be identical.
 - (c) Variable declarations can appear almost anywhere in the body of a C++ program.
 - (d) The default case is required in the **switch** selection structure.
 - (e) An array may store many different types of values.
 - (f) Both protected and public members of a super-class are visible to the derived subclass.
 - (g) Protected members of a super-class are visible to friends of the sub-class.
 - (h) If class A is a friend of class B, and class B is a friend of class C, then class A is a friend of class C.
 - (i) A class may have more than one constructor functions.
 - (j) A subclass inherits only the public members of its super-class.
- (ii) Write a C++ statement(s) to accomplish each of the following [10 marks]
- (a) Declare an double precision pointer y, set it to point to some arbitrary memory location, and initialize the value of this location to be 7.4.
 - (b) Print the value float X in a field of 15 characters with precision of 3.
 - (c) Sum the multiples of 5 between 0 and 100 using a **for** loop.
 - (d) Declare an array of 5 integers called **scores** and initialize all its components to 0.
 - (e) Determine and print the largest value contained in array **w** of floats.

Question 2 – 20 marks

- (i) Briefly describe the five (5) basic concepts of object-oriented programming? [5 marks]
- (ii) Explain how each of the concepts described in (i) above is implemented in C++? [15 marks]

Question 3 – 20 marks

- (i) Explain the following concepts, giving examples where possible?
- (a) Function Prototype *1 mark*
 - (b) Function Signature *1 mark*
 - (c) Function Overloading *1 mark*
 - (d) Friend function *1 mark*
 - (e) Access member function *1 mark*
- (ii)
- (a) What is a destructor function? *2 marks*
 - (b) How and when is a destructor function called? *2 marks*
 - (c) A class may not have more than one destructor functions. Why is this the case? *2 marks*
- (iii)
- (a) What is a polymorphic function? *2 mark*
 - (b) Explain two (2) cases (concepts) which give rise to polymorphic functions. *3 marks*
 - (c) What is function template? *2 marks*
 - (d) Do you see any relationship between a function templates and polymorphism? *2 marks*

Question 4 – 20 marks

- (i) Using a function template, write a function **Swap** that takes two variables as arguments and swaps the values of the variables. [5 marks]
- (ii) Using a template, write a function **IndexOfSmallest** (*A, first, last*) that takes three arguments. The first argument *A* is an array, and the second and third arguments are integer index values. The function must return the index of the smallest value in the array in the index range *first* to *last*. [7marks]
- (iii) Using the functions **Swap** and **IndexOfSmallest** above, write a function **SelectionSort** (*A, first, last*) that takes an array and sorts the elements in the array **recursively**. The first argument is the name of the array, *first* is the index of first element in the array and *last* is the index of last element in the array. The pseudocode for recursive selection sort is as follows, assuming elements are sorted in ascending order.

SelectionSort (A, first, last)

Begin

Find smallest element in A[first..Last] and place it in its correct positions, A[first].

Find n smallest element in A[first+1..last] and place it in its correct position, A[first+1].

Repeat until all elements are sorted

End;

[8 marks]

Question 5– 20 marks

Write a C++ function **DoubleSpace** (*ifstream InFile, ofstream OutFile*) that takes an input stream, *InFile*, as input and copies all the contents of *InFile* to an output stream, *OutFile*. The lines in *OutFile* must be double-spaced. That is there must be at least one blank line between any consecutive lines in *OutFile*. The figure below shows a sample *InFile* and the expected double spaced *OutFile*.

InFile – Single Spaced

Good Hamlet,
Cast thy nited color
And look like a friend to Denmark

OutFile – double spaced

Good Hamlet,

Cast thy nited color

And look like a friend to Denmark

[20 marks]

Question 6– 20 marks

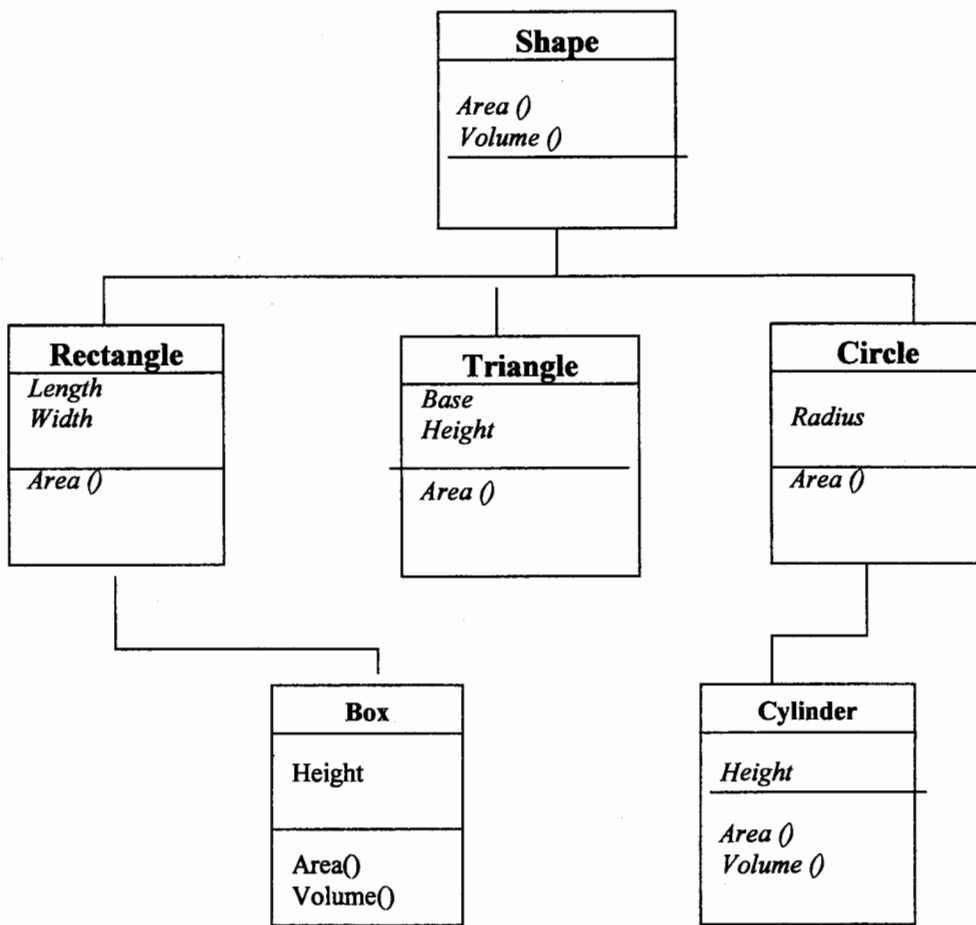
Create a class called **IntegerSet**. Each object of class **IntegerSet** can hold integers in the range 0 through 100. A set is represented internally as an array of ones and zeros. Array element **a[i]** is 1 if *i* is in the set. Array element **a[j]** is 0 if *j* is not in the set. The default constructor initializes a set to an empty set. A set is empty if all its array elements are initialized to 0.

Implement the following operations on a set:

- **Insert (e)** – adds a new element *e* into the set.
- **Remove (e)** – remove element *e* from the set.
- **IsMember (e)** – returns true if element *e* is in the set and false otherwise.
- **Print ()** – display all elements of the set on standard output.

Question 7– 20 marks

Consider the following class hierarchy for shapes



Using C++ notation implement the above class hierarchy. Your implementation must be as close as possible to the above design. You may add necessary operations, but all the attributes have been specified. [20 marks]

The following formulas are useful.

$$\begin{aligned} \text{VolumeOfCylinder} &= \text{BaseCircleArea} * \text{Height}; \\ \text{AreaOfCylinder} &= 2 * \text{BaseCircleArea} + 2 * \text{Pi} * \text{RadiusofBaseCircleArea} * \text{Height}. \end{aligned}$$

$$\begin{aligned} \text{VolumeOfBox} &= \text{BaseRectangleArea} * \text{Height}; \\ \text{AreaOfBox} &= 2 * \text{BaseRectangleArea} + 2 * \text{Height} * \text{Length} + 2 * \text{Height} * \text{Width}; \end{aligned}$$