King’s College London

UNIVERSITY OF LONDON

This paper is part of an examination of the College counting towards the award of a degree. Examinations are governed by the College Regulations under the Authority of the Academic Board.

M.Sci. EXAMINATION

CP/4735 The C++ programming language for physicists

SUMMER 1998

Time allowed: **TWO HOURS**

Candidates must answer any TWO questions. No credit will be given for attempting a further question.

The approximate mark for each part of a question is indicated in square brackets.

Good answers to questions will include plans and explanations in addition to sections of C++ code.

TURN OVER WHEN INSTRUCTED

Answer TWO questions

1) What is meant by any three of the following terms in C++:
inheritance, polymorphism, overloading, constructor, hiding?

[6 marks]

Explain the differences between procedural and object oriented programming (OOP) and why OOP is considered by many to be a superior approach.

[4 marks]

Write short sections of C++ code to illustrate how the `+` operator could be overloaded to add vectors of length 3. Pay special attention to the accessibility of your variables.

[10 marks]

2) Declare a class called `complex` which deals with complex numbers. It should include the overloaded operators `+`, `-` and the function `conjugate` which calculates the complex conjugate of the complex number.

[10 marks]

Define another class of lines in the complex plane, using the class `complex` in a suitable way. It should include a function, which returns the length of the line.

[6 marks]

Write code to show how these classes would be used.

[4 marks]

SEE NEXT PAGE

3) This C++ header file defines a class `list` for a linked list:

```
struct element { int num; element *next; } ;

class list
{
public:
      list(){ start = end = new element;};
      ~list();
      void Append(int i);
      void IteratorStart(){ iter_ptr = start;}
      int IteratorOK(){ return iter_ptr != end;}
      int Iterator();
private:
      element *start, *end, *iter_ptr;
};

list :: ~list()
{
element *p;
while (start != end)
{
      p = start;
      start = start->next;
      delete p;
}
delete start;
}

void list ::Append(int i)
{
end->num = i;
end = end->next = new element;
}

int list ::Iterator()
{
int i = iter_ptr->num;
iter_ptr->next;
return i;
}
```

Explain in detail what each function does and how this linked list works.

[16 marks]

Write a main function which would read a specified number of integers into a list of this type.

[4 marks]

FINAL PAGE