Section A

Question A1.

(a) List three reasons why code reuse based on libraries and components has reached its limits and suggest one way in which code generation may help this situation.

[5 marks]

(b) What is meant by *horizontal* and *vertical* transformations with respect to code generation?

[5 marks]

(c) What is meant by a *compositional* code generator? Provide one benefit of using a compositional code generator.

[5 marks]

(d) Explain the difference between a function and a macro in the C programming language.

[5 marks]

(e) What is a meta-program? Give two commonly used examples of meta-programs.

[5 marks]
Section B

Question B1.

(a) Describe the three different kinds of hygiene ascribed to C style macros. [6 marks]

(b) Although the C/C++ programming languages support multi-dimensional arrays, it is more efficient to use a single dimensional array. Write C style macros to implement the following using only a single dimensional array:

- **MAKE2DARRAY**(T,N,M,a) declares a 2 dimensional array with entries of type T, the number of rows and columns is N and M respectively, and the name of the array is a.

- **SET2D**(a,i,j,v) writes the value v in to the 2 dimensional array named a at position (i, j)

- **GET2D**(a,i,j) gives the value of the (i, j)th entry of the 2 dimensional array a. [10 marks]

(c) Without explicitly declaring an array, use your macros from (b) above to implement the following C style macros:

- **MAKE3DARRAY**(T,N,M,K,a) declares a 3 dimensional array with entries of type T, with size N × M × K, and the name of the array is a.

- **SET3D**(a,i,j,k,v) writes the value v in to 3 dimensional array named a at position (i, j, k)

- **GET3D**(a,i,j,k) gives the value of the (i, j, k)th entry of the 3 dimensional array a. [5 marks]

(d) Explain in what way the macros above may fail to be hygienic. [4 marks]

TURN OVER
Question B2.

(a) Explain the difference between a joinpoint and a pointcut descriptor. [4 marks]

(b) Explain the difference between the withincode and cflow pointcut descriptors? [4 marks]

(c) Describe in words the methods matched by the following signature patterns:

(i) * boolean java..*.is*(..)

(ii) !static * java..*.Get*(*,*)

[6 marks]

(d) Can there be methods that are matched by both signatures given in Question B2 (c) above? If yes, construct an example, if not, explain why not. [4 marks]

(e) Consider the following Java code

```java
class A {
    void m() { System.out.println("Hello"); }
}

class B extends A {
    void m() { super.m(); System.out.println("World"); }
}

class C extends B {
}
```

Write an AspectJ aspect that supresses the printing of “Hello” whenever a call to method m() is made via an instance of class C. For example, `new B().m()` should print “Hello World” (with appropriate line breaks) but `new C().m()` should just print “World”. [7 marks]
Question B3.

(a) Explain the quotation/unquotation mechanism in Meta-AspectJ.  
[6 marks]

(b) Explain the semantics of the \texttt{infer} keyword in Meta-AspectJ.  
[4 marks]

(c) Suppose there are different client applications that call a Calendar Server application using a method named \texttt{find} that returns an array of \texttt{Event} objects.

Most clients use the method \texttt{find} by specifying a date as follows:

\begin{verbatim}
find(int year, int month, int day)
\end{verbatim}

However, others use an \textbf{alternative} specification of the date in a different order:

\begin{verbatim}
find(int day, int month, int year).
\end{verbatim}

Write a Meta-AspectJ method called \texttt{generateFind} that accepts a \texttt{String[]} containing an array of string class names of clients that use the alternative date specification. The method should generate code to allow the different clients to call \texttt{find} using their own preferred orders for the parameters.

[Hint: generate an aspect that catches calls to \texttt{find}, checks the type of the calling object, and rearranges the arguments appropriately. ]
[15 marks]
Question B4.

(a) What do the terms *partial evaluation* and *staging* mean in the context of C++ template meta-programming? [6 marks]

(b) Describe which compile time errors may arise during template instantiation. [4 marks]

(c) Consider the following templates that represent a List data structure.

```cpp
struct Nil
{
    enum { head = ERROR };
    typedef Nil Tail;
};
template <int head_, class Tail_ = Nil>
struct Cons
{
    enum { head = head_ };
    typedef Tail_ Tail;
};
```

Write a templated function that accepts two such list structures and statically *zips* them to create a single new list. A zip of two lists is a list whose elements alternate between elements of the two lists. For example, the zip of [4; 5; 6] and [1; 2; 3] is [4; 1; 5; 2; 6; 3]. You may assume that the given lists are of the same length. [15 marks]