Answer four out of five questions.

Each question is worth 15 marks. This examination is worth 60%.
The courseworks were worth 40%.
Examination and courseworks total 100%.

Calculators may NOT be used.
1. a) List three benefits of automatic code generation, and explain why code generators provide these.  
   (3 marks)

   b) Name three different code generation paradigms.  
   (3 marks)

   c) Explain the difference between horizontal and vertical transformations.  
   (4 marks)

   d) Explain what a generator framework is. What advantages do they provide for building code generators?  
   (5 marks)

2. a) Explain when a macro is called hygienic. List the different levels of macro hygiene that exist, and explain what assurance they provide, and how this can be achieved.  
   (8 marks)

   b) Write a C-style macro DCL_FIELD that generates Java-Code for a private field declaration and its public getter/setter methods, given the name and type of the field.  
   (7 marks)
3. a) Explain the difference between a joinpoint and a pointcut descriptor.  
(4 marks)

b) Describe in words what is matched by the following method signatures:
   (i) `javax..Tag*.get*(..)`
   (ii) `*Info javax.servlet.jsp.tagext.*.get*()`  
(4 marks)

c) Can there be methods that are matched by both signatures given in question 3 b) above? If yes, construct an example, if not, explain why not.  
(3 marks)

d) What, if any, is the difference between these four patterns?
   (i) `javax..*Model`
   (ii) `javax..*Model+`
   (iii) `javax..*Model`
   (iv) `javax..*Model()`  
(4 marks)
4. a) Write an aspect `CacheFib` that caches the method `fib` defined below. The aspect should store the results of calls to `fib` in a table and return these if `fib` is called recursively with the same argument. Your cache does not need to be persistent between different top-level (i.e., non-recursive) calls to `fib`. Briefly explain your code.

(10 marks)

```java
public class Example {
    static int fib(int n) {
        if (n <= 1)
            return n;
        else
            return fib(n-1) + fib(n-2);
    }
}
```

b) How would you modify your `CacheFib` aspect so that it can cache arbitrary, user-specified methods with one `int`-argument and an `int`-result? You should assume that the user maintains a String set `cachedMethods` in a class `CacheControl` that contains the names of all such methods that should be cached.

(5 marks)
5.

a) Consider the four template definitions below:

(I) \( \text{template <class T1, class T2> class P {...};} \)
(II) \( \text{template <class T1, class T2> class P<T1*, T2*> {...};} \)
(III) \( \text{template <> class P<bool, bool> {...};} \)
(IV) \( \text{template <class T> class P<T, T> {...};} \)

For each of the following five declarations describe which specialization is selected, and why, or describe why the declaration is causing an error.

(i) \( \text{P<bool, bool> a;} \)
(ii) \( \text{P<int*, float*> b;} \)
(iii) \( \text{P<bool, int> c;} \)
(iv) \( \text{P<int*, int*> d;} \)
(v) \( \text{P<int, int> e;} \)

(5 marks)

b) Consider the class MyVector given below. Write a template metaprogram that generates a specialized instance of the dot-product of two elements of the class MyVector. The dot-product of two vectors \( x \) and \( y \) is defined as \( x \cdot y = \sum x[i] \cdot y[i] \).

\[
\begin{align*}
\text{template<class T, int N> class MyVector } & \ \
\text{public: } & \\
& \text{T operator[](int i) const} \\
& \{ \text{return data[i]; } \} \\
\text{private: } & \\
& \text{T data[N];} \\
\end{align*}
\]

(10 marks)

END OF PAPER