UNIVERSITY OF SOUTHAMPTON

SEMIESTER 2 EXAMINATION 2014 - 2015

AUTOMATED CODE GENERATION

DURATION 90 MINS (1.5 Hours)

This paper contains 5 questions

Answer FOUR questions. Each question is worth 15 marks. You should spend around 22.5 minutes on each question.

An outline marking scheme is shown in brackets to the right of each question.

This examination is worth 60% of the overall module mark. The coursework is worth 40%.

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

6 page examination paper.
Question 1.

(a) Explain the difference between code generation and compilation. Give one example for each of the two techniques.

[4 marks]

(b) Explain the difference between horizontal and vertical transformation in code generation. Provide an example of a vertical transformation.

[4 marks]

(c) List three different types of non-code artefacts that a code generator can produce.

[3 marks]

(d) Describe the difference between horizontal and vertical scaling in software libraries.

[4 marks]
Question 2.

(a) Explain macro hygiene. [3 marks]

(b) Refactor the following C program using macros in order to avoid duplication of code usage patterns.

```c
#include <stdio.h>

int main() {
    float f1[10];
    for (int i = 0; i < 10; i++)
        f1[i] = f1[i] * f1[i];
    float f2[10];
    for (int i = 0; i < 10; i++)
        f2[i] = f2[i] + f2[i];
    double d1[20];
    for (int i = 0; i < 20; i++)
        d1[i] = d1[i] * d1[i];
    double d2[20];
    for (int i = 0; i < 20; i++)
        d2[i] = d2[i] + d2[i];
}
```

[9 marks]

(c) Decide whether the following C program compiles. If it does, write the output. Otherwise, explain why it does not compile.

```c
#include <stdio.h>
#define POW3(N) N*POW2(N)
#define POW2(N) N*N

int main() {
    int a = 2;
    int b = POW2(a);
    int c = POW3(a);
    printf("a = %d, b = %d, c = %d", a, b, c);
}
```

[3 marks]
Question 3.

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

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

(i) * java..io.Buf*.get*(*)
(ii) public InputStream java..*.get*Stream(boolean) [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 signature patterns?

(i) gui.*Element()
(ii) gui..*Element*
(iii) gui..*Element
(iv) gui..*Element+ [4 marks]
Question 4.

(a) Explain the quotation/unquotation mechanism in Meta-AspectJ. 

[7 marks]

(b) Explain in words the output produced by the following method:

```java
void genCountingLogger(String cn, Stm Logger) {
    infer aspectCode =
        [ public aspect #[cn + ?_CountingLogger"] {
            int count = 0;
            before : call(* #cn.*(..)) {
                count++;
                #Logger;
            }
        } ];
    aspectCode.unparse();
}
```

[8 marks]
Question 5.

(a) Consider the five template definitions below:

a) template <class T1, class T2>
   class C<T1*, T2> ...;

b) template <class T1, class T2>
   class C<T1, T2*> ...;

c) template <class T>
   class C<T, T> ...;

d) template <class T>
   class C<T*, T*> ...;

e) template <>
   class C<char*, char*> ...;

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

(i) C<char, int> a;

(ii) C<int*, int*> b;

(iii) C<char, int*> c;

(iv) C<int*, char*> d;

(v) C<int, int> e;

[5 marks]

(b) Using template metaprogramming for recursion and for the if template, write a C++ program that computes \( \min(M^N, 10) \), where \( M \) and \( N \) are integers such that \( M > 0 \) and \( N \geq 0 \).

[10 marks]