SEMESTER 2 EXAMINATION 2013 - 2014

FROM ASPECT ORIENTED PROGRAMMING TO PROGRAM GENERATION

DURATION 90 MINS (1.5 Hours)

This paper contains 5 questions

Answer FOUR questions. Each question is worth 15 marks.

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.
Question 1.

(a) Explain the difference between model-based and code-based generation. 

[4 marks]

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

[3 marks]

(c) Give related examples of a Model, a Meta-Model, and a Meta-Meta-Model. 

[4 marks]

(d) Describe the library scaling problem. 

[4 marks]
Question 2.

(a) Provide an example of a C program with a macro which does not guarantee structural integrity. [3 marks]

(b) Provide an example of a C program with a macro which does not guarantee referential integrity. [3 marks]

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

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

[6 marks]

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

```c
#define FACT(N) (N > 1) ? (N*FACT(N-1)) : 1
int main() {
    int k = FACT(5);
    printf("%d",k);
}
```

[3 marks]
Question 3.

(a) Explain the difference between the static and dynamic pointcut designators. Give an example for each of the two kinds. [3 marks]

(b) Describe in words the joinpoints matched by the following pointcut designators:
   
   (i) call(public * java.lang.*ss.is*())
   
   (ii) call(* String.*(..)) && args(*, int)
   
   (iii) call(java.io..new(..))

   [6 marks]

(c) Consider the following class:

```java
public class A {
    private int k = 1;

    public A() { }

    public void doOne() {
        if (k-- != 0)
            doOne();
        else
            doTwo();
    }

    public void doTwo() { return; }

    public static void main(String[] args) {
        A someA = new A();
        someA.doOne();
    }

```
Explain what happens and write the output of the program when each of the following advices is applied.

Advice 1:

```java
before(): call(* A.*(..)) && cflow(call(* doOne())) {
   System.out.println(thisJoinPoint.getSignature());
}
```

Advice 2:

```java
before(): call(* A.*(..)) && cflow(call(* doTwo())) {
   System.out.println(thisJoinPoint.getSignature());
}
```

Advice 3:

```java
before(): call(* A.*(..)) && cflowbelow(call(* doTwo())) {
   System.out.println(thisJoinPoint.getSignature());
}
```

[6 marks]
Question 4.

(a) Describe what a domain-specific aspect language is and discuss an example. [3 marks]

(b) Describe the keyword `infer` of Meta-AspectJ. [3 marks]

(c) Describe advantages and limitations of SafeGen over Meta-AspectJ. [3 marks]

(d) Consider the following method in an Meta-AspectJ program:

```java
void generateTrivialLogging(String classNm) {
    infer aspectCode =
    '
        package MyPackage;
        aspect #[classNm + "Aspect"] {
            before : call(* #classNm.*(..))
                { System.out.println("Method called"); }
        }
    ];
    System.out.println(aspectCode.unparse());
}
```

Write an equivalent method in **plain Java** which produces the same output. [6 marks]
Question 5.

(a) Consider the following fragment of C++ program.

```cpp
template <class T1, class T2>
class MyClass {
   // Template A
};
template <class T>
class MyClass<T,T> {
   // Template B
};
template <class T>
class MyClass<T,int> {
   // Template C
};
template <class T1, class T2>
class MyClass<T1*,T2*> {
   // Template D
};
```

For each of the following declarations say which template will be used by the declaration or why the compiler issues an error.

(i) MyClass<int*,int*> m1;
(ii) MyClass<float,int> m2;
(iii) MyClass<float,float> m3;
(iv) MyClass<int,float> m4;
(v) MyClass<int*,float*> m5;
(vi) MyClass<int,int> m6;

[6 marks]
(b) Recall that the sequence of Fibonacci numbers, denoted $F_n$, is defined by the recurrence relation:

\[ F_n = F_{n-1} + F_{n-2}, \quad \text{with} \quad F_1 = 1 \quad \text{and} \quad F_0 = 0. \]

Define the generalised sequence of Fibonacci numbers, denoted by $F_n^\alpha$, as follows:

\[ F_n^\alpha = \alpha(F_{n-1}^\alpha + F_{n-2}^\alpha), \quad \text{with} \quad F_1^\alpha = 1, F_0^\alpha = 0, \text{for any} \quad \alpha = 1, 2, \ldots. \]

Write a C++ program that computes $F_5^2$ and $F_{10}^3$ at compile time using template meta-programming. [9 marks]