6G6Z1110 – Programming Languages: Principles and Design
Dr Ryan Cunningham: room E142 (ryan.cunningham@mmu.ac.uk)

Q1. Semantic Analyser [25 marks]

There are 3 parts to this question: **a**, **b**, and **c**. Please answer any two questions.

Please complete the **Lab B - Semantic Analysis Coursework** and upload your **semantics.py** file (see assessment area on moodle).
In a comment at the top of your file, you must state which two tasks you have attempted.

a) Method Invocation

```
import printf, printf;
void foo(int x, int y) {
  return 0;
}
int a = food(1, -2.0, 5);
```

Write solutions which identify all semantic errors (**hint:** rules 4, 6, 7, 26) in the above code and write code in your *CoffeeTreeVisitor* class to detect the errors.

b) Arithmetic and Logic

```
if (true && !1) {
    return true;
}
```

Write solutions which identify all semantic errors (**hint:** rules 8, 9, 11, 22) in the above code and write code in your *CoffeeTreeVisitor* class to detect the errors.

c) For Loops

```
bool n = 10;
for (i in [:n:2.]) {
  return;
}
continue;
```

Write solutions which identify all semantic errors (**hint:** rules 8, 10, 13, 27) in the above code and write code in your *CoffeeTreeVisitor* class to detect the errors.

Q2. Code Generator [25 marks]

There are 3 parts to this question: **a**, **b**, and **c**. Please answer **any two** questions.

Please complete the **Lab B - CodeGen Coursework** and upload your **codegen.py** file (see assessment area on moodle).
In a comment at the top of your file, **you must** state which two tasks you have attempted.

a) Expressions

Write solutions which generate the correct assembly code and program output for the Coffee programs below:

i) Arithmetic:

```
int a, b;
a = 2 + 3 * 4;
b = 5 - a % 10;
return -(a + b);
```

b) Methods

Write solutions which generate the correct assembly code and program output for the Coffee programs below:

i) Extra Args:

```
int sum(int a, int b, int c, int d, int e, int f, int g) {
  return a + b + c + d + e + f + g;
}
return sum(1, 2, 3, 4, 5, 6, 7);
```

c) Loops

Write solutions which generate the correct assembly code and program output for the Coffee programs below:

i) Limit / Step:

```
int a = 0;
for (i in [1:10:2]) {
  a = a + i;
}
return a;
```

Marking Grid

For each of the 2 questions above, a percentage of the available marks will be awarded following the marking scheme below:

Criteria	**Bad Fail (0-29)**	**Fail (30-39)**	**3rd (40-49)**	**2:2 (50-59)**	**2:1 (60-69)**	**1st (70-79)**	**Exceptional 1st (80+)**
Demonstrate a high degree of professionalism: correctness of solution; completeness of solution; demonstration of specific compiler concepts; clarity of code presentation (including comments).	Code does not compile, or so limited as to provide no clear path to a solution; there is no, or almost no evidence of knowledge of compiler concepts; code presentation is inadequate.	Code has limited correctness; evidence of knowledge of compiler concepts is very limited; code presentation is poor.	Code performs incorrectly in a significant number of tests; evidence of knowledge of compiler concepts is limited; code presentation is adequate.	Code performs correctly against a majority of testing, but some significant shortcomings are not recognised (using comments); some knowledge of compiler concepts is demonstrated; code presentation is reasonable.	Code performs correctly against a majority of testing, but there are some significant shortcomings that have been recognised, or there some minor shortcomings that are not recognised (using comments); evidence of knowledge of compiler concepts is demonstrated; code presentation is good.	Code performs correctly against significant majority of testing, and any failures are noted as known shortcomings (using comments); evidence of knowledge of compiler concepts is demonstrated; code presentation is excellent.	Code performs correctly against all testing; evidence of knowledge of compiler concepts is demonstrated; minor shortcomings are recognised (using comments) code presentation is exceptional.