CIS 606 Final Fall 2000

Instructions: This is an open book, open note, 110 minute exam. There are 2 questions worth a total of 100 points. You must answer both of the questions. Put your name on all pieces of paper that you turn in. Show all of your work, write legibly and good luck.

1) (30 points) The question is about the architecture of compilers and the JOOS compiler in particular.

a) **(5 pts)** Define *compiler phase* and *compiler pass.*

b) **(15 pts)** Identify the phases and at least 5 passes in your project compiler. Of your 5 passes, at most 2 of them can come from the same phase. Give a one sentence description of what each of phase and pass does.

c) **(10 pts)** Describe at least 3 mechanisms for passing information between phases and passes in the JOOS compiler.

2) (70 points) You are to add a switch statement to JOOS and enhance your compiler appropriately.

Syntactically, a switch looks like the following:

```
switch (<expr>) {
  case <literal> : <stmt>; break;
  case <literal> : <stmt>
  default : <stmt>
}
```

where `break` statements optionally define the end of a `case` and `default` defines the case in which the `<expr>` value is not equal to any of the `<literal>`s.

Semantically, the `<expr>` must be an integer expression and the `<literal>` must be integer constants.

In your answers below you are to refer to the following example

```
switch (x+1) {
  case 0 : y=0; break
  case 1 :
  case 3 :
  case 5 : y=1; break
  case 2 : y=0; break
  default : y=-1;
}
```

Describe the changes you would make to your compiler in the following areas:

a) **(10 points)** Extend the scanner to handle the new lexical elements.

b) **(10 points)** Extend the parser to handle the new syntactic elements.

c) **(10 points)** Extend the AST for switch statements. Illustrate it for the above example.

d) **(10 points)** Describe the type checking rule for the new AST nodes.

e) **(15 points)** Describe a general JVM byte-code template for this statement. Illustrate how the template is expanded for the above example.

f) **(10 points)** Describe a peephole optimization to improve the performance of a switch with empty cases. Illustrate how it improves the byte-codes for the example generated in the previous question.

g) **(5 points)** Give at least one reason why one cannot easily implement switch statements as syntactic sugar.