COL226: Programming Languages

Mon 14 Feb 2022 **MinorQ3** 5+15 (+5 for PwD) minutes Max marks 10

Instructions:

1. Download the paper and write your name and entry number in the designated space on top and *do not forget to sign the honour statement below.*
2. Answer the question(s). *Answers will be judged for correctness, efficiency and elegance.*
4. If there are minor mistakes in the question, correct them explicitly and answer the question accordingly. If the question is totally wrong, give adequate reasons why it is wrong with detailed counter-examples, if necessary.
4. Scan the paper with your completed answer.
5. Upload it on Gradescope 2102-COL226 page within the given time. *Make sure the first page with your name, entry no and signature is also the first page of your uploaded file*
6. Late submissions (within 2 minutes of submission deadline) on the portal will attract a penalty of 10% of the total marks allotted to the paper for each minute of delay and 20% for each minute of delay thereafter.
7. Email submissions after the closing of the portal will not be evaluated (You get a 0).
8. Uploads without the first page details (including signature) may be awarded 0 marks.

I abide by the Honour code that I have signed on my admission to IIT Delhi. I have neither given any help to anybody nor received any help from anybody nor from any site or other sources in solving the question(s) in this paper.

Signature: **Date:**

[5+5=10 marks]

In the bad old days, LISP and APL programming languages used a dynamic scoping mechanism to resolve non-local references. In practical terms, this meant that there were no static links in the activation records, and all non-local references were accessed through the dynamic link only. Hence given a variable (identifier) that may have been declared in multiple scopes, any reference to it referred to the "declaration of that identifier in the most recent scope occurrence in the calling chain".

Now consider the following self-evident program written in a self-evident syntax. The words `program` and `procedure` are keywords and parameter-passing is by value.

```
program mypgm ;
   var x:integer;

   procedure setx (n:integer);
   { x := n}

   procedure showx();
   { print (x)}

   procedure first();
   { set(1); showx()}

   procedure second();
      var x:integer;
   { setx(2); showx()}

// Main pgm
{ setx(0); first(); showx(); second(); showx()}
```

1. What values does this program print if the language uses *static scoping* with *dynamic memory allocation*? Explain using a diagram of the runtime stack.
2. What values does this program print if the language uses *dynamic scoping* with *dynamic memory allocation*? Explain using a diagram of the runtime stack.