1. (i) What should the pre-condition \( P \) be in each of the following correctness statements for the statement to be an instance of Hoare's axiom scheme? All variables are of type \( \text{int} \).
   (a) \( P \{ x = y + 2*z; \} \; w*x >= z + 23 \)
   \[ w \times (y + 2 \times z) \geq z + 23 \]
   (b) \( P \{ x = y + z; \} \) \( \exists x (x = 0; \; x < 2*y) \; u*x + 2 \leq v + z \)
   \[ \exists x (x=0; \; x<2*y) \; u \times x + 2 \leq v + z \ (\text{no subst.}) \]
   (c) \( P \{ x = y + z; \} \) \( \exists z (z=0; \; z < u) \; x+z + 3 \geq y*v \)
   \[ \exists z (z=0; \; z < u) \; y+z + a + 3 \geq y \times v \]

(ii) Verify the validity of the below correctness statement by adding all the intermediate assertions, that is, give the proof tableau. All variables are of type \( \text{int} \). Clearly state any mathematical facts and inference rules used.

```
ASSERT(x == 4 && y == 7)
  x = x - y;
  y = y + x;
  x = y - x;
ASSERT(x == 7 && y == 4)
  ASSERT(x == 4 && y == 7)  // \&\& is commutative
  ASSERT(y == 7 && y + x - y == 4)
  x = x - y;
  ASSERT(y + x - x == 7 && y + x == 4)
  y = y + x;
  ASSERT(y - x == 7 && y == 4)
  x = y - x;
ASSERT(x == 7 && y == 4)
```
2. (i) (2 marks) Using left-factoring and/or elimination of left-recursion give grammars equivalent to the below two grammars where the immediate problems preventing use of recursive-descent parsing have been removed. Capital letters denote variables and the set of terminals is \{a, b, c, d\}.

(a) \[ S \rightarrow Sab \mid Sc \mid dc \mid \epsilon \]
\[ S \rightarrow dS' \mid S' \]
\[ S' \rightarrow abS' \mid cS' \mid \epsilon \]

(b) \[ S \rightarrow abSa \mid abcSa \mid bdSc \mid bc \mid cc \]
\[ S \rightarrow abS' \mid bS'' \mid cc \]
\[ S' \rightarrow aS \mid cSa \]
\[ S'' \rightarrow dSc \mid \epsilon \]

(ii) (3 marks) Consider the language \( L = \{a^{2i}b^{2k}c^i \mid k \geq 0, i \geq 1\} \)
Draw a pushdown automaton that recognizes the language \( L \).
3. For each question, circle one answer. If you circle more than one answer, it will be considered a wrong answer. If in doubt, it is to your advantage to make a guess.

(i) What is the language generated by the grammar \( S \to bScc | bS | \epsilon \)

(a) \( \{ b^i c^{3k} | 0 \leq i \leq k \} \)
(b) \( \{ b^i c^{3k} | 0 \leq k \leq i \} \)
(c) \( \{ b^i c^k | 0 \leq 3k \leq i \} \)
(d) None of the above.

(ii) Which of the following statements is true:

(a) There exists a language \( L \) recognized by a nondeterministic pushdown automaton such that \( L \) cannot be recognized by any deterministic pushdown automaton.
(b) There exists a language \( L \) recognized by a deterministic pushdown automaton such that \( L \) cannot be recognized by any nondeterministic pushdown automaton.
(c) There exists a regular language \( L \) such that \( L \) cannot be recognized by any deterministic pushdown automaton.
(d) The above three statements are all false.

(iii) Consider a context-free grammar that has productions \( S \to \alpha | \beta \)
where \( \alpha \) derives the empty string. Which of the following conditions always prevents the use of recursive-descent parsing with this grammar:

(a) \( \text{first}(\beta) \cap \text{first}(\beta) \neq \emptyset \)
(b) \( \text{follow}(\beta) \cap \text{first}(\beta) \neq \emptyset \)
(c) \( \text{follow}(\beta) \cup \text{first}(\alpha) \neq \emptyset \)
(d) None of the conditions prevents the use of recursive-descent parsing.

(iv) Determine \( \text{follow}(S) \) for the grammar \( (S, A \text{ are nonterminals, } S \text{ is the start nonterminal}) \):
\[
S \to aSAa | \epsilon, \quad A \to aAcd | \epsilon
\]

(a) \( \text{follow}(S) = \{ a, b \} \)
(b) \( \text{follow}(S) = \{ a, b, c \} \)
(c) \( \text{follow}(S) = \{ a, b, \text{EOS} \} \)
(d) \( \text{follow}(S) = \{ a, b, c, \text{EOS} \} \)
(e) None of the above.

(v) Let \( P \) and \( Q \) be any assertions. The following holds always:

(a) \( P \) is stronger than \( Q \) or \( Q \) is stronger than \( P \)
(b) \( P \bot Q \) is stronger than \( P \)
(c) \( P \text{ and } Q \) is stronger than \( Q \)
(d) None of the above.

(vi) Consider the correctness statement: \( Q \{ z = z+10; \} \) \( z > 0 \)
where \( z \) has type integer. The correctness statement is valid when \( Q \) is the assertion:

(a) \( z < 0 \)
(b) \( z <= 0 \)
(c) false
(d) The correctness statement is invalid in all above three cases.
4. Are the below languages $A$ and $B$ over alphabet $\Sigma = \{a, b, c, d, f\}$ context-free or non-context-free?

- If a language is context-free, give a context-free grammar that defines it.
- If a language is not context-free, prove using the pumping lemma that it is not context-free.

(i) $A = \{a^{2i}b^{3k}c^{k+1}d^{i+1} \mid i \geq 1, k \geq 1\}$

(ii) $B = \{b^i c^k d^i \mid k \geq 0\} \cdot \{d^{2m}f^m \mid m \geq 0\}$

(Here "\cdot" is the concatenation of languages.)

i) Grammar for $A$:

$$S \rightarrow a^2 S d \mid a^2 X d^2$$

$$X \rightarrow b^3 X c \mid b^3 c^2$$

ii) $B$ is not context-free: Assume $B$ is CF and let $p$ be the constant given by the pumping lemma. Choose $s = b^p c^p d^p \in B$ (the string should not have a's.) By the PL we can write $s = uvwx$, the pumping lemma conditions.

a) If $v$ or $x$ contains more than one type of symbol, then $uv^2wx^2y \notin B$ and is not in $B$.

In the following $v$ and $x$ each contain at most one type of symbol.

b) If $v$ or $x$ contains $c$'s, then $uv^0w^0x^0y = uvx$ has fewer $c$'s than $b$'s or fewer $c$'s than $d$'s, and is not in $B$.

c) The remaining possibility is that $v$ and $x$ contain only $b$'s and $d$'s. Now $uv^2wx^2y$ has either (i) more $b$'s than $c$'s, or (ii) more $d$'s than $c$'s and no $f$'s. In both cases $uv^2wx^2y$ is not in $B$. All cases lead to a contradiction.