Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

EXAMEN FINAL

SEG 2506 - Construction de logiciels

Date: 20 avril 2022 **Professeur:** Aziz Oukaira

Durée: 180 minutes **Session:** Hiver 2022

Consignes :

- Cet examen est à livre ouvert.
- Une seule feuille de papier double face est autorisée.
- Répondez à toutes les questions, en commençant par la plus facile pour vous.
- En cas de doute sur le sens d'une question, énoncez clairement toute supposition que vous faites.
- Veuillez lire la déclaration d'intégrité académique et mettez vos initiales.
- **Toute tentative de plagiat ou de collaboration au plagiat entraînera la note 0.**

Nom de l'étudiant (e) : _____ **Numéro étudiant :** _____

L'examen est pour un total de 100 points.

Questions	Pointage Maximum	Notes
1	30	
2	30	
3	20	
4	20	

Je comprends l'importance de l'intégrité professionnelle dans ma formation et ma future carrière d'ingénieur. Je certifie par la présente que j'effectuerai tous les travaux relatifs à l'examen SEG2506 entièrement par moi-même, sans aide extérieure ni utilisation de sources d'information non autorisées. En outre, je ne fournirai aucune assistance à d'autres personnes. Je reconnais avoir lu et compris les instructions ci-dessus, et je m'engage à les respecter pendant cet examen.

Vos initiales : .....

Bonne chance!

QUESTION #1 (30 points)

Chaque question à choix multiple ne possède qu'une seule réponse correcte.

1. (2 points) Les méthodes Agile :
   a. Rendre obligatoire la création de conceptions complètes et très détaillées avant le codage.
   b. Mettre l'accent sur la documentation.
   c. Rendre obligatoire le développement d'un prototype pour mieux comprendre les besoins avant de commencer la conception.
   d. Rendent obligatoire l'utilisation d'itérations dans le cadre du processus de développement.
   e. Toutes ces réponses.
2. (2 points) Le manifeste agile stipule que les développeurs de logiciels devraient :
   a. S'assurer de la qualité de l'application produite en la déployant d'abord dans un environnement de test avant de la livrer au client.
   b. Se concentrer sur la création de modèles structurels et comportementaux détaillés qui couvrent tous les aspects du système avant de commencer le processus de codage.
   c. Maintenir une interaction absolument minimale avec le client (pour ne pas lui faire perdre de temps).
   d. S'attendre à ce que les exigences du système changent et donc concevoir le système en fonction de ces changements.
   e. Aucune de ces réponses
3. (2 points) Les exigences non fonctionnelles décrivent :
   a. Formellement, à l'aide de diagrammes de cas d'utilisation, ce que le système est censé faire.
   b. De manière informelle, à l'aide d'énoncés faciles à comprendre, ce que le système est censé faire.
   c. Les attributs du système, tels que la sécurité, la fiabilité, la maintenabilité, la scalabilité et la facilité d'utilisation.
   d. Les attributs du projet, tels que le nombre de développeurs et le niveau d'expertise requis pour mener à bien le développement.
   e. Aucune de ces réponses
4. (2 points) Les modèles structurels définissent :
   a. L’architecture statique d'une application logicielle.
   b. Le comportement d'une application logicielle.
   c. Les propriétés concurrentes d'une application logicielle.
   d. La synchronisation entre les différents processus d'une application logicielle.
   e. Aucune de ces réponses.
5. (2 points) L'analyse lexicale consiste à :
   a. L'identification de phrases composées de mots appartenant à une langue.
   b. L'identification de mots constitués de caractères appartenant à un alphabet.
   c. L'évaluation du sens attaché aux phrases.
   d. L'examen de la structure d'une phrase.
   e. Aucune de ces réponses.

6. (2 points) La fermeture (closure) d'un alphabet est :
    a. L'ensemble de toutes les expressions régulières qui dénotent un langage formel.
    b. L'ensemble des chaînes de caractères qui contiennent deux caractères ou plus de l'alphabet.
    c. L'ensemble de toutes les chaînes possibles composées à partir d'un alphabet, y compris la chaîne vide.
    d. L'ensemble de toutes les chaînes possibles composées à partir d'un alphabet, à l'exception de la chaîne vide.
    e. Aucune de ces réponses.
7. (2 points) Sélectionnez l'affirmation la plus appropriée :
    a. La plupart des erreurs de compilation sont détectées pendant la phase d'analyse lexicale.
    b. La plupart des erreurs de compilation sont détectées pendant la phase d'analyse syntaxique.
    c. La plupart des erreurs de compilation sont détectées pendant la phase d'optimisation du code.
    d. La plupart des erreurs de compilation sont détectées pendant l'exécution du programme.
    e. Aucune de ces réponses.
8. (2 points) Un langage qui peut être généré par une grammaire contextuelle est un:
    a. Langage formel.
    b. Langage naturel.
    c. Langage sans contexte.
    d. Langage de programmation.
    e. Langage parlé.
9. (2 points) Étant donné un non-terminal A dans la grammaire G, FOLLOW(A) est :
    a. L'ensemble de tous les terminaux qui apparaissent en premier dans toutes les chaînes possibles qui peuvent être dérivées en une ou plusieurs étapes de A.
    b. L'ensemble de tous les terminaux qui apparaissent en dernier dans toutes les productions qui contiennent A du côté gauche.
    c. L'ensemble de tous les terminaux qui apparaissent en dernier dans toutes les productions qui contiennent A du côté droit.
    d. L'ensemble de tous les terminaux qui apparaissent après A dans toutes les chaînes de caractères qui peuvent être générées par la grammaire G.
    e. Aucune de ces réponses.
10. (2 points) Nous pouvons conclure que deux grammaires sont équivalentes :
    a. Si leurs ensembles FIRST et FOLLOW sont les mêmes.
    b. Si elles ont le même nombre de terminaux et de non-terminaux.
    c. Si elles définissent le même nombre de règles.
    d. Si elles génèrent le même langage.
    e. Aucune de ces réponses.
11. (2 points) L'analyseur syntaxique LL(1) détecte une erreur de syntaxe si :
    a. Le sommet de la pile de dérivation est un non-terminal qui ne correspond pas à la tête de la liste de flux de tokens.
    b. Le sommet de la pile de dérivation est un terminal qui ne correspond pas à la tête de la liste de flux de tokens.
    c. Le sommet de la pile de dérivation est un non-terminal et la combinaison du sommet de la pile de dérivation et de la tête de la liste de flux de tokens correspond à une cellule vide dans la table d'analyse syntaxique.
    d. Options (a) et (b).
    e. Options (b) et (c).

12. (2 points) Une grammaire compatible LL(1) :
    a. Ne peut pas être ambiguë.
    b. Doit être factorisée à gauche.
    c. Ne peut pas être récursive à gauche.
    d. Tout ce qui précède.
    e. Aucune de ces réponses.
13. (2 points) Étant donné la grammaire G et le langage L(G), nous pouvons conclure que G est ambiguë lorsque :
    a. En utilisant les règles de G, nous ne pouvons pas générer avec succès toutes les chaînes de caractères qui appartiennent à L(G).
    b. En utilisant les règles de G, nous pouvons générer deux arbres d'analyse différents pour une seule chaîne dans L(G).
    c. La grammaire n'est pas compatible avec LL(1).
    d. La grammaire nécessite l'élimination de la récursion gauche.
    e. Aucune de ces réponses.
14. (2 points) En général, pour les environnements multi-threads, deux types de mécanismes de synchronisation existent :
    a. La synchronisation de coopération et de compétition.
    b. Synchronisation d'exécution simultanée et séquentielle.
    c. Passage de messages et rendez-vous asynchrones.
    d. Toutes ces réponses.
    e. Aucune de ces réponses.
15. (2 points) Laquelle des affirmations suivantes est vraie dans le contexte de la communication interprocessus (IPC) :
    a. La communication synchrone exige que le récepteur attende que l'expéditeur soit prêt à envoyer.
    b. La communication entre deux processus locaux (exécutés sur le même ordinateur) se fait toujours par le biais de la mémoire partagée.
    c. La communication entre des processus distants (exécutés sur des ordinateurs différents) est toujours réalisée par le biais de la mémoire partagée.
    d. La communication asynchrone ne peut pas inclure d'appels bloquants.
    e. Aucune de ces réponses.

QUESTION #2 (30 points)

1. (10 points) Est-ce que la grammaire suivante est ambigüe ? Expliquez en quelques mots pourquoi elle l'est ou ne l'est pas.

   A → a A c | a A | b

2. (10 points) Prouvez que la grammaire contextuelle suivante est ambiguë :

   S → ABCD

   A → ae | af | ag | ah

   B → b | ε

   C → hcd | bcd | cd

   D → e | f | g | h

Rappel: Pour prouver que la grammaire est ambiguë, nous devons trouver au moins une chaîne qui peut être générée à l'aide de deux arbres d'analyse différents (parsage).

3. (10 points) Nous considérons la grammaire suivante:

   S → S a B | A

   A → d A | c

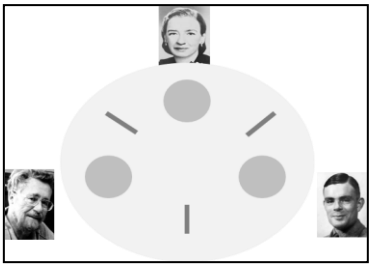
   B → b | A

Lesquelles des phrases suivantes sont générées par cette grammaire ?

1. cab
2. c
3. cabb
4. caab
5. ccaabb
6. ccaab
7. ddcac

QUESTION #3 (20 points)

Trois informaticiens (Grace Hopper, Alan Turing et Edsger Dijkstra) sont assis autour de la table du dîner et réfléchissent à des problèmes complexes (voir la jolie figure). Sur la table, il y a trois assiettes de nourriture et trois baguettes. Les baguettes sont placées de manière que chaque scientifique puisse avoir accès à deux baguettes, une à sa gauche et une autre à sa droite.



Lorsqu'un informaticien a faim, il fait une pause dans sa réflexion et saisit les deux baguettes (une à sa gauche et une autre à sa droite) **exactement au même instant** et mange. Lorsqu'il/elle a fini de manger, il/elle pose les baguettes et reprend sa réflexion. Les informaticiens alternent à l'infini entre les états de penser et de manger. Il est évident qu'une baguette ne peut pas être utilisée par deux informaticiens simultanément. Par conséquent, si un informaticien veut manger, mais qu'une ou deux des baguettes nécessaires sont utilisées, il doit attendre que les baguettes soient disponibles.

Concevez un réseau de Petri qui modélise le problème des informaticiens dîneurs. À l'aide d'étiquettes, décris ce que représente chaque place du réseau de Petri.

QUESTION #4 (20 points)

Nous considérons quatre processus arrivent dans l'ordre P1, P2, P3 et P4.

- Temps de burst P1 : **36**
- Temps de burst P2 : **23**
- Temps de burst P3 : **3**
- Temps de burst P4 : **8**

1. (10 points) Dessinez le diagramme de Gantt qui montre l'ordre dans lequel l'ordonnanceur place les processus sur l'unité centrale en utilisant l'algorithme **Round Robin** avec un quantum de temps de 10 unités.
2. (10 points) Calculez le **temps d'attente** moyen et le **temps d'achèvement** moyen pour chaque processus.

Fin de l'examen.