

S.O.007 PROGRAMMATION DU RÉSEAU
1. A propos de la discipline

Faculté	Ordinateurs, informatique et microélectronique				
Département	Génie logiciel et automatisation				
Cycle d'étude	Licence, premier cycle				
Programme d'études	Technologies de l'information				
Année d'études	Semestre	Type d'évaluation	Catégorie formative	Catégorie d'optionnalité	Credite ECTS
3ème année (enseignement à temps plein)	6	E	S-Discipline de specialitate	O - unité d'enseignement obligatoire	4
Année IV (temps partiel)	8				

2. Durée totale estimée

Nombre total d'heures de cours		Dont			
		Heures d'ouverture de l'auditorium		Travailler individuellement	
		Curs	Laboratoires	Étude du matériel théorique	Préparation des candidatures
Enseignement à temps plein	120	30	30	30	30
Enseignement à temps partiel	120	12	12	40	56

3. Conditions d'accès à la discipline

Selon le programme d'études	Architecture informatique, Analyse et modélisation des systèmes d'information, Programmation C++, Analyse, programmation et conception orientées objet, Systèmes d'exploitation
Selon le compétences	Application de langages de programmation, d'environnements de modélisation et de développement, de méthodologies pour la création de logiciels

4. Conditions du processus éducatif pour les

Cours	Un projecteur et un ordinateur sont nécessaires pour présenter le matériel théorique en classe.
Laboratoires	Les étudiants rédigeront les rapports requis par les lignes directrices méthodologiques. Date limite de remise des travaux de laboratoire : une semaine après leur achèvement.

5. Compétences spécifiques acquises

Compétence professionnelle	CP2. Conception et développement de l'application Connaissances: • Programmes/modules logiciels appropriés. • Composants matériels, outils et architectures. • Technologies de pointe. • Langages de programmation • Développement rapide d'applications (RAD). • Technologie et langages de modélisation technique. • Questions de sécurité. Compétences: • Explique et communique au client les informations relatives à la conception et au développement de l'application. • Effectuer et évaluer les résultats des tests par rapport aux spécifications du produit. • Appliquer les architectures logicielles et/ou matérielles appropriées. • Gérer et assurer un niveau élevé de qualité et de cohésion. • Utiliser des modèles de données. • Effectuer et évaluer les résultats des tests dans l'environnement du client ou de la cible. Normes minimales de performance pour l'évaluation des compétences: • Faire preuve de créativité pour développer des applications et sélectionner les options techniques appropriées. • Participer à d'autres activités de développement. • Optimiser le développement, la maintenance et les performances des applications en utilisant des modèles de conception et en réutilisant des solutions testées.
-----------------------------------	--

CP3. Intégration des composants

Connaissances:

- Composants matériels/logiciels/modules, qu'ils soient anciens, existants ou nouveaux.
- Impact de l'intégration d'un nouveau système sur l'organisation ou le système existant.
- Techniques d'interfaçage entre modules, systèmes et composants.
- Techniques de test d'intégration.
- Outils de développement (par exemple, environnement de développement, gestion des modifications, contrôle des modifications et accès au code source).
- Bonnes pratiques de conception.

Compétences:

- Mesure les performances du système avant, pendant et après son intégration.
- Vérifie que les capacités et l'efficacité des systèmes intégrés sont conformes aux spécifications..

Normes minimales de performance pour l'évaluation des compétences:

- Tenez compte de vos propres actions et de celles des autres dans le processus d'intégration.
- Respecter les normes et procédures de contrôle appropriées pour maintenir l'intégrité de la fonctionnalité et la fiabilité globale du système.

CP5. Mise en œuvre de solutions

Connaissances:

- Techniques d'analyse des performances.
- Techniques de gestion des problèmes (fonctionnement, performance, compatibilité).
- Impact de la mise en œuvre/déploiement sur l'architecture existante.
- Technologies et normes à utiliser lors de la mise en œuvre/déploiement.

Compétences:

- Organiser le processus de mise en œuvre et les activités de lancement du produit.
- Organiser et planifier les activités de bêta-test et de test de la solution dans son environnement opérationnel final.
- Configurer les composants à n'importe quel niveau pour assurer une interopérabilité globale correcte.
- Identifie et fait appel à l'expertise nécessaire pour résoudre les problèmes d'interopérabilité.

Normes minimales de performance pour l'évaluation des compétences:

- Tenir compte de ses propres actions et de celles des autres pour proposer des solutions et engager la communication et la collaboration avec les parties prenantes.
- Apporter son expertise pour influencer, par le biais de conseils et d'assistance, l'élaboration de solutions.

CP7. Ingénierie des systèmes

Connaissances:

- Composants matériels, outils et architectures matérielles.
- Conception fonctionnelle et technique.
- Technologies de pointe.
- Langages de programmation.
- Les bases de la sécurité de l'information.

Compétences:

- Explique et communique au client les informations relatives à la conception et au développement.
- Libère et évalue les résultats des essais par rapport aux spécifications du produit.
- Appliquer les architectures logicielles et/ou matérielles appropriées.
- Conçoit et développe l'architecture matérielle, les interfaces utilisateur, les composants logiciels professionnels et les composants logiciels intégrés.
- Gérer et assurer des niveaux élevés de cohésion et de qualité dans le développement de logiciels complexes.
- Utilise des modèles de données.
- Appliquer les modèles et/ou processus de développement appropriés pour développer de manière efficace et productive..

Normes minimales de performance pour l'évaluation des compétences:

- Tire parti de ses connaissances spécialisées et de sa compréhension approfondie de l'infrastructure TIC et du processus de gestion des problèmes pour identifier les défaillances et les résoudre avec le moins de perturbations possible.
- Prendre des décisions éclairées dans des situations émotionnellement tendues sur les actions appropriées nécessaires pour minimiser l'impact sur l'entreprise..
- Identifier rapidement les composants défectueux, choisir les solutions de réparation, de remplacement ou de reconfiguration..

Compétences transversales	CT3. Développement personnel et professionnel Reconnaît la nécessité de l'apprentissage tout au long de la vie en utilisant efficacement les ressources et les techniques d'apprentissage pour le développement personnel et professionnel.
----------------------------------	---

5. Les objectifs

Objectif général	Le cours vise à donner aux étudiants un aperçu des techniques de programmation en réseau, à développer une compréhension de l'évolution du domaine et des références futures, à renforcer les connaissances de base et à les doter des compétences nécessaires pour appliquer ces connaissances en pratique dans la création d'un système de transfert de données efficace..
Objectifs spécifiques	• Comprendre l'architecture et les fonctionnalités des réseaux informatiques. • Développer une bonne connaissance des protocoles de communication de réseau tels que TCP/IP. • Être capable de programmer des applications client-serveur et de comprendre les concepts associés. • Être capable d'identifier et de résoudre les problèmes de sécurité dans les réseaux informatiques. • Connaître les technologies et les tendances actuelles dans le domaine de la programmation de réseaux, telles que les réseaux définis par logiciel (SDN) ou les technologies en nuage. • Avoir la capacité de travailler en équipe pour concevoir et mettre en œuvre des solutions de réseau complexes. • Comprendre l'importance de la documentation et des normes dans la programmation en réseau. • Appliquer les connaissances acquises dans le développement d'un projet pratique de programmation de réseau.

6. Contenu du sujet

Thèmes des activités d'enseignement	Nombre d'heures	
	éducation à temps plein	l'enseignement à temps partiel
Thèmes des cours		
T1. Programmation de réseaux informatiques. Principes fondamentaux	2	1
T2. Programmation de réseaux à l'aide du protocole TCP	4	1
T3. Programmation en réseau à l'aide du protocole UDP	4	1
T4. Programmation du serveur de réseau avec plusieurs connexions parallèles	4	2
T5. Mise en œuvre d'un protocole personnalisé	4	1
T6. Transfert de données en réseau : sérialisation des objets	2	1
T7. Communiquer avec le DNS	2	1
T8. Programmation de réseaux à l'aide des protocoles HTTP, SMTP et FTP	4	2
T9. Programmation de réseaux à l'aide de langages de programmation modernes	4	2
Total des cours:	30	12
Thèmes des travaux de laboratoire		
LL1. Application de chat utilisant TCP	8	3
LL2. Application de chat utilisant UDP	8	3
LL3. Application client DNS	4	1
LL4. Application de transfert de courrier électronique	4	2
LL5. Application client HTTP	4	2
LL6. Application client NTP	2	1
Total des travaux de laboratoire:	30	12

7. Références bibliographiques

Principal	1. M. Bancila, Modern C++ Programming Cookbook: Master C++ core language and standard library features, with over 100 recipes, updated to C++20, 2nd Edition, Packt Publishing (September 11, 2020) 2. S. Burns, Hands-On Network Programming with C# and .NET Core: Build robust network applications with C# and .NET Core, Packt Publishing; 1st edition (March 29, 2019) 3. A. Tanenbaum, Rețele de calculatoare (ediția a patra), Byblos, Tg.Mureș, 2003 (Э. Таненбаум, Компьютерные сети. Питер, 2003) 4. Lupșa Radu-Lucian, Retele de calculatoare, Casa Cărții de Știință, 2008, ISBN: 978-973-133-377-9, http://www.cs.ubbcluj.ro/~rlupsa/works/retele.pdf 5. Семенов Ю. А., Телекоммуникационные технологии, http://saturn.itep.ru 6. Joseph Albahari, Threading in C# (online), http://www.albahari.com/threading/ Frăsinaru Cristian, Curs practic de Java, (Capitolul 13. Programare în rețea), Matrix Rom, 2005, http://thor.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf
Suplimentare	1. Bass L., Clements P., Kazman R. Software Architecture in Practice, Addison Wesley, 2003 2. A. S. Tanenbaum, M. van Steen, Distributed Systems. Principles and paradigms, Prentice Hall, 2007. 3. Карпов Л. Е., Архитектура распределенных систем программного обеспечения. Учебное пособие, М.: МАКС Пресс, МГУ, 2007. 4. V. Kumar, A. Grama, A. Gupta, G. Karypis, Introduction to Parallel Computing, Benjamin-Cummings, 2003.

8. Évaluer

Périodique		Actuel	Étude individuelle	Projet/thèse	Examen
EP 1	EP 2				
Enseignement à temps plein					
15%	15%	15%	15%		40%
Norme de performance minimale Présence et activité aux cours magistraux et aux travaux de laboratoire; Obtenir une note minimale de "5" pour chacune des attestations et des travaux de laboratoire ; Démonstration, dans l'épreuve de l'examen final, de la connaissance des processus et technologies de base appliqués au développement d'applications en réseau.					

9. Critères d'évaluation

Activer	Composant évaluer	Méthode d'évaluation, Critères d'évaluation	Part dans la note finale de l'activité	Poids dans l'évaluation du sujet
Enseignement à temps plein				
L'évaluation périodique I	Contenu théorique, thèmes 1-5	Test sur MOODLE	100%	15%
L'évaluation périodique II	Contenu théorique, thèmes 6-9	Test sur MOODLE	100%	15%
Évaluation actuelle	Activité pratique	Discussions en laboratoire	50%	15%
		Dossier complété par des rapports pour chaque travail de laboratoire	50%	
L'étude individuelle	Recherche sur le thème	Conférence publique/présentation/discours	100%	15%
L'évaluation finale	Contenu théorique et et pratique	Test sur MOODLE	100%	40%