**TP 1**

**(Travail d’équipe : deux étudiants)**

Distribué en classe le jeudi 2 juin 2011

**Date de remise : le jeudi 16 juin 2011**

**Exercices de programmation**

**Objectif pédagogique**

Ce travail pratique porte sur Javascript et les bases de la programmation. L’étudiant(e) doit démontrer ses aptitudes à écrire des programmes en Javascript qui utilisent les concepts de base de la programmation comme les structures de répétition et de condition, l’utilisation de variables, de fonctions et des objets fournis par Javascript. Les concepts importants à évaluer sont la capacité de résoudre un problème grâce à la programmation et l’aptitude à respecter la syntaxe d’un langage de programmation.

**Ce qu’il faut accomplir**

Vous développerez quatre pages différentes avec du code JavaScript qui permettront d’effectuer différentes tâches.

1. **1ère page : Calcul du coût d’un voyage en Europe (4 points)**

Cette première application demandera quelques informations à l’usager soit : le nombre de journées passées en voyage, le coût par jour en hôtel, le coût par jour en nourriture ainsi qu’un coût total pour acheter des souvenirs, tous les montants étant en Euros. L’application calculera ensuite le total en dollars canadiens en utilisant un facteur de conversion de 1,56. Toutes les informations seront affichées sur la page même.

1. **2ème page : Détermination du palindrome (5 points)**

La deuxième application servira à déterminer si un mot ou une phrase est un palindrome. Un palindrome est une suite de lettres qui, lorsque renversée, donne le même résultat qu’auparavant. Par exemple, Laval est un palindrome. Elu par cette crapule est aussi un palindrome. Il faut ignorer les espaces, les virgules, les points et la casse lorsque l’on veut déterminer si une phrase est un palindrome. L’usager entrera une phrase et vous devrez vérifier si cette phrase est un palindrome et lui donner le résultat.

1. **3ème page : Traitement d’infos sur une classe d’étudiants (12 points)**

Vous devez créer une série de fonctions qui utilisent une classe d’étudiants. Cette classe doit respecter un certain format, qui est expliqué ci-dessous.

La classe est un tableau dans lequel se trouvent des étudiants.

Un étudiant est une chaîne de caractère divisée par des virgules et qui se lit comme suit : « Prénom Nom, Age, NoteTP, NoteExam ».

**Pour avoir une classe pour tester vos fonctions, créez une page qui contient au départ ce script :**

<script language="javascript">

var classe = new Array(5);

classe[0] = "Réjean Thibodeau,40,80,90";

classe[1] = "Éric Tremblay,21,70,65";

classe[2] = "Annie Boucher,30,80,80";

classe[3] = "Ken Smith,30,90,70;

classe[4] = "Jennifer Harvey,12,90,90";

</script>

Les fonctions que vous devez créer et tester sont :

1. Une fonction qui prend en paramètres un indice numérique et qui calcule la note finale de cet étudiant. La note finale est calculée en attribuant 60% de la pondération à l’examen et 40% au TP. (6 points)
2. Une fonction qui calcule l’âge moyen de la classe. (3 points)
3. Une fonction qui retourne le nom d’un étudiant pris au hasard dans la classe. (3 points)

Toutes ces fonctions doivent utiliser la classe définie globalement dans votre script (elle n’est pas envoyée en paramètres).

Toutes ces fonctions doivent s’adapter à la taille de la classe, c’est-à-dire que vous pouvez modifier la taille de votre tableau classe (rajouter des étudiants) et que les fonctions marcheront encore suite à cette modification.

Vous devez démontrer le fonctionnement de ces fonctions par l’affiche du résultat des appels dans des document.write().

1. **4ème page : Jeu de roche-papier-ciseau (5 points) Difficile**

La dernière application est un jeu de roche papier ciseau entre l’usager et l’ordinateur. Tout d’abord, l’ordinateur doit générer une valeur au hasard entre roche, papier et ciseau. Ensuite, il demande à l’usager d’entrer une valeur, soit roche papier ou ciseau. Si le choix de l’usager est le même que celui de l’ordinateur, on recommence le jeu jusqu’à ce que les deux choix soient différents. Si les choix sont différents, on détermine le gagnant de la manière habituelle, soit roche gagne sur ciseau, ciseau gagne sur papier, et papier gagne sur roche.

Il y aura aussi 4 points (sur 30) d’accordés pour les pratiques de programmation. Produisez donc un code clair, bien indenté, commenté, et modulaire (définissez des fonctions lorsqu’approprié), et une interface fonctionnelle et facile à utiliser. Pour avoir ces 4 points, les entrées de l’usager doivent aussi être validées pour le format et le contenu.

**Le total compte pour 20% de votre résultat final.**

**Ce qu’il faut remettre**

Remettez le projet final dans un dossier identifié par vos noms:

**Nom1\_nom2**

et déposez ce dernier dans LEA avant minuit à la date d’échéance.