

# CORRECTION

<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1
<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2
<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3
<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4
<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5
<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6
<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7
<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8
<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9

← codez votre numéro d'étudiant  
ci-contre, et inscrivez votre nom et prénom  
ci-dessous.

Nom : .....

Prénom : .....

## LIFAP4 Conception et développement d'applications : Examen (CC mi-parcours) du 14 février 2019 (90 minutes)

Aucun document autorisé, sans calculatrice, sans téléphone

**Attention : noircir complètement à l'encre la ou les cases cochées (une croix ne suffit pas).** Les questions marquées d'un trèfle peuvent avoir 0, 1 ou plusieurs choix corrects. Ce sera le cas de toutes les questions dans ce sujet.

**Question 1 ♣** Avec un 'git commit' quelle option permet d'ajouter un commentaire ?

- ☐ On ne peut pas inclure de commentaire directement.
- ☐ git commit -c "le commentaire"
- ☒ git commit -m "le commentaire"
- ☐ git commit -co "le commentaire"

**Question 2 ♣** Ce code

```
0: class Truc
1: {
2:     int a;
3:     int& getA() const { return a; }
4: }
```

- ☐ provoquera une erreur avec un débogueur.
- ☐ provoquera un 'segmentation fault (core dump)'.
- ☒ provoquera une erreur de compilation.
- ☐ provoquera une erreur avec Valgrind.
- ☐ ne provoquera pas d'erreur.

**Question 3 ♣** Cochez les affirmations vraies à propos d'un diagramme de Gantt.

- ☐ Un diagramme de Gantt indique les pénalités en cas de retard dans le développement.
- ☒ La colonne de gauche du diagramme énumère toutes les tâches à effectuer.
- ☒ La ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.).
- ☒ Un diagramme de Gantt permet de visualiser dans le temps les diverses tâches composant un projet.

**Question 4 ♣** Avec Git, le dépôt local

- ☐ se trouve sur le serveur Git.
- ☒ se trouve dans le répertoire '.git'.
- ☐ est l'ensemble des fichiers sur mon disque dur sur lesquels je travaille directement.
- ☒ contient tout l'historique des versions.

**Question 5 ♣** Avec un débogueur je peux

- ☒ mettre des points d'arrêts.
- ☐ lancer une série de tests dans un script.
- ☒ inspecter des variables au cours de l'exécution.
- ☐ valider des correctifs.
- ☐ retrouver l'historique des exécutions.
- ☐ corriger les erreurs de compilation.
- ☐ obtenir des statistiques d'utilisation de chaque fonction.
- ☒ comprendre ce que fait réellement le programme.

**Question 6 ♣** La balise `\brief` sert à

- ☒ documenter une fonction.
- ☒ documenter une structure/class.
- ☒ donner une description courte.

**Question 7 ♣** Avec Git, la commande `'git commit'`

- ☐ permet d'envoyer le dépôt local vers le dépôt distant.
- ☐ permet d'envoyer les changements des fichiers de travail vers le dépôt local et vers le dépôt distant.
- ☒ permet d'envoyer uniquement les changements des fichiers de travail vers le dépôt local.

**Question 8 ♣** Nous avons besoin de passer une structure lourde en place mémoire à une procédure. La bonne manière de faire est ?

```
0: struct Gros {
1:     int tab[32000];
2: }
3: int main()
4: {
5:     Gros g;
6:     float m = moyenne(g);
7: }
```

- ☐ `float moyenne(Gros g)`
- ☒ `float moyenne(const Gros& g)`
- ☐ `float moyenne(Gros& g)`

**Question 9 ♣** Pour gérer toutes les notes de LIFAP4 nous écrivons la classe suivante :

```
0: class LIFAP4
1: {
2: public:
3:     LIFAP4() : nb_notes(0), taille_notes(0) {}
4:     int sauver(const string& filename) const;
5:     void ajouterUneNote(int n) { CODE }
6: private:
7:     int *notes;
8:     int nb_notes, taille_notes;
9: };
```

- ☐ Un destructeur ne sera pas utile dans cette classe.
- ☒ Ligne 7 : `notes` est un tableau.
- ☒ Ligne 7 : `notes` est un pointeur.
- ☒ Ligne 3 : il manque une/des initialisation(s).
- ☒ Ligne 4 : le `'const'` à la fin signifie que la fonction ne modifie pas les données membres de la classe.
- ☐ Ligne 4 : le `'const'` à la fin signifie que la fonction renvoie un entier constant.
- ☒ ligne 5 : l'instruction `CODE` fera une (re)allocation dynamique si le tableau est plein.

## CORRECTION

**Question 10 ♣** La directive ci-dessous

```
#ifndef _CAMION_H_
#define _CAMION_H_
...
#endif
```

- ☒ se trouve dans le fichier 'Camion.h'.
- ☐ provoquera une erreur d'exécution.
- ☐ provoquera une erreur de compilation si elle n'est pas respectée.
- ☒ sert à éviter les redéfinitions de types/classes/fonctions.
- ☐ se trouve dans le fichier 'Voiture.h'.
- ☐ se trouve dans le fichier 'Camion.cpp'.

**Question 11 ♣** Cochez les affirmations appartenant à la méthode de conception AGILE

- ☐ On ne revient pas sur un choix quoi qu'il arrive.
- ☐ Les processus complexes sont importants pour résoudre des problèmes complexes.
- ☒ Bâissez le projet autour de personnes motivées.
- ☒ Un logiciel fonctionnel est la meilleure unité de mesure de la progression du projet.
- ☒ La méthode la plus efficace pour transmettre l'information est une conversation en face à face.
- ☐ Les processus AGILE promeuvent un rythme de développement tranquille mais sur.

**Question 12 ♣** La fonction 'SDL\_RenderCopy' de SDL2

- ☐ copie une surface sur la fenêtre principale ou tout autre surface cible.
- ☒ permet de sélectionner une partie de la texture source et une taille du rectangle destination.
- ☒ copie une texture sur la fenêtre principale ou tout autre texture cible.

**Question 13 ♣** La fonction 'SDL\_RenderPresent' de SDL2

- ☒ met à jour la fenêtre principale en affichant tout les affichages faits depuis l'appel précédent.
- ☐ recopie le 2e buffer d'affichage dans le buffer visible.
- ☒ bascule les deux buffers d'affichages.

**Question 14 ♣** Avec Git, le dépôt distant

- ☐ est l'ensemble des fichiers sur mon disque sur lequel je travaille.
- ☐ se trouve dans le répertoire '.git'.
- ☒ se trouve sur le serveur Git.

**Question 15 ♣** Ce code

```
int *a = new int[10];
a[0] = 12;
delete [] a;
if (a[0]==12) cout<<"Vrai";
```

- ☒ provoquera une erreur avec Valgrind.
- ☐ est un code correcte.
- ☒ peut provoquer un 'segmentation fault (core dump)'.
- ☐ provoquera une erreur de compilation.

**Question 16 ♣** SDL permet facilement

- ☒ de récupérer les touches claviers enfoncées par l'utilisateur.
- ☐ de faire saisir à l'utilisateur du texte.
- ☒ d'afficher des images.
- ☐ d'ajouter des menus à une application.
- ☒ de jouer des sons.

**Question 17 ♣** Même si un programme marche, est-ce que Valgrind peut servir quand même?

- ☒ Oui
- ☐ Non

## CORRECTION

**Question 18 ♣** Je veux supprimer un fichier de mon historique de Git, :

- ☒ je ne peux pas.
- ☐ j'utilise la commande 'git rm'.
- ☐ j'utilise la commande 'rm' de mon terminal.

**Question 19 ♣** Le cahier des charges

- ☐ définit le nombre de niveau qu'aura le jeu.
- ☒ indique les modalités d'exécution pour la réalisation avec les éléments à livrer (des démos).
- ☒ est un contrat entre le client et le prestataire de service.
- ☒ définit exhaustivement les spécifications de base de l'application.
- ☐ est un cahier à spirales.
- ☐ indique les bureaux où les développeurs seront assis.
- ☐ propose des cycles accélérés de conception.

**Question 20 ♣** Dans un Makefile, il y a des règles

- ☒ comportant une tabulation à gauche sur la 2e ligne.
- ☐ comportant les dépendances à gauche sur la 1ere ligne.
- ☒ comportant les dépendances à droite sur la 1ere ligne.
- ☐ comportant des espaces à gauche sur la 2e ligne.
- ☒ comportant la règle de production sur la 2e ligne.
- ☒ comportant la cible à gauche sur la 1ere ligne.
- ☒ sur 2 lignes.
- ☐ comportant la cible à droite sur la 1ere ligne.

**Question 21 ♣** Avec Git, pour récupérer une copie de travail dans le répertoire courant, la commande est

- ☐ git download http://forge.univ-lyon1.fr/monprojet
- ☐ git checkout http://forge.univ-lyon1.fr/monprojet
- ☒ git clone http://forge.univ-lyon1.fr/monprojet

**Question 22 ♣** La commande 'g++ -Wall -ggdb -c main.cpp -I/usr/include'

- ☐ crée un exécutable.
- ☐ '-I/usr/include' indique un chemin vers les bibliothèques (.a, .so ou .dll).
- ☒ '-Wall' provoque l'affichage de tous les messages de prévention (ambiguïtés, type cast, oublis, etc.).
- ☒ effectue la phase de compilation.
- ☐ '-ggdb' indique d'exécuter le programme avec gdb.
- ☒ '-I/usr/include' indique un chemin vers les fichiers d'entête (.h).
- ☒ '-ggdb' indique d'inclure les informations sur le programme pour le débogueur.
- ☐ effectue la phase d'édition de lien.
- ☒ '-c' demande la production de fichiers objets (.o).

**Question 23 ♣** Ce code

`assert(a!=0);`

- ☒ arrête l'application si 'a' vaut 0.
- ☐ indique au compilateur de tester si la valeur de a est différente de 0.
- ☒ s'assure à l'exécution que la variable 'a' est différente de 0.
- ☐ indique au débogueur de tester si la valeur de a est différente de 0.
- ☐ indique au débogueur de mettre une valeur différente de 0 dans a.

## CORRECTION

**Question 24 ♣** Cochez les actions qui sont exécutées classiquement par une boucle d'événement dans une IHM

- ☐ Envoie des événements à un autre programme.
- ☒ Affiche l'application.
- ☒ Exécute des actions de manière automatique.
- ☐ Pose des questions à l'utilisateur.
- ☒ Récupère et traite des événements du clavier.
- ☐ Sauvegarde toutes les données dans un fichier d'historique.
- ☒ Récupère et traite des événements de la souris.
- ☐ Ouvre une fenêtre.
- ☒ Efface l'écran.

**Question 25 ♣** Pour comprendre, inspecter et déboguer la gestion de la mémoire d'un programme je peux utiliser

- ☐ Git.
- ☐ Doxygen.
- ☒ des 'cout/printf' dans les constructeurs et destructeurs.
- ☐ les erreurs de g++.
- ☒ un débogueur.
- ☒ Valgrind.

**Question 26 ♣** Avec Git, la commande 'git log'

- ☒ affiche l'historique des commentaires entrés lors des commit.
- ☐ enregistre mes changements dans un journal de log.
- ☐ enregistre mes commentaires dans un journal de log.

**Question 27 ♣** Cochez les règles d'écriture de code importantes.

- ☒ Variables, fonctions et classes avec des noms claires et explicites.
- ☒ Code indenté.
- ☐ Le code doit avoir le moins de ligne possible.
- ☒ Des lignes ne dépassant pas 80 colonnes.
- ☐ Plus de commentaire que de code.
- ☐ Toujours indenter avec des espaces et non des tabulations.

**Question 28 ♣** Mon programme ne fait pas ce qu'il devrait faire, mais ne plante pas, quelle solution ai-je pour le corriger ?

- ☒ Comprendre ce qu'il fait réellement avec un débogueur.
- ☐ De corriger les erreurs de compilation.
- ☒ Le tester avec Valgrind.
- ☒ Comprendre ce qu'il fait réellement avec des affichages 'cout/printf'.

**Question 29 ♣** Segmentation fault (core dumped) signifie

- ☒ une erreur de segmentation.
- ☐ une erreur de Doxygen.
- ☒ une erreur à l'exécution.
- ☐ une erreur à la compilation.
- ☒ que cette application a causé une erreur due à la mémoire.
- ☒ que le système d'exploitation a stoppé cette application.

**Question 30 ♣** Avec Git, la commande git pull

- ☒ permet de récupérer les changements du dépôt distant dans le dépôt local et de mettre à jour les fichiers de travail
- ☐ permet de répercuter uniquement les changements du dépôt local dans les fichiers de travail
- ☐ permet de récupérer uniquement les changements du dépôt distant dans le dépôt local

## CORRECTION

**Question 31 ♣** Avec Git, la copie de travail

- ☐ se trouve sur le serveur Git.
- ☐ contient uniquement des fichiers gérés par Git.
- ☐ se trouve dans le répertoire '.git'.
- ☒ est l'ensemble des fichiers sur mon disque dur sur lesquels je travaille directement.

**Question 32 ♣** Avec Git quelle est l'opération inverse de 'git add' ?

- ☐ rm
- ☒ git rm fichier
- ☐ git reset HEAD

**Question 33 ♣** Avec Git, quelle commande demande un accès réseau ?

- ☒ git push
- ☒ git pull
- ☐ git commit

**Question 34 ♣** Avec Git, la commande 'git push'

- ☒ permet d'envoyer le dépôt local dans le dépôt distant.
- ☐ permet d'envoyer uniquement les changements des fichiers de travail dans le dépôt local.
- ☐ permet d'envoyer les changements des fichiers de travail dans le dépôt local et dans le dépôt distant.

**Question 35 ♣** Ce code

```
0:  class Truc
1:  {
2:      int a;
3:      void faire() const { a=12; }
4:  }
```

- ☐ ne provoquera pas d'erreur.
- ☐ provoquera un 'segmentation fault (core dump)'.
- ☐ provoquera une erreur avec Valgrind.
- ☒ provoquera une erreur de compilation.
- ☐ provoquera une erreur avec un débogueur.

**Question 36 ♣** Ce code

```
0:  class Truc
1:  {
2:      int a;
3:      int getA() const { return a; }
4:  }
```

- ☐ provoquera une erreur de compilation.
- ☒ ne provoquera pas d'erreur.
- ☐ provoquera une erreur avec un débogueur.
- ☐ provoquera une erreur avec Valgrind.
- ☐ provoquera un 'segmentation fault (core dump)'.

**Question 37 ♣** Doxygen analyse les commentaires commençant par

- ☒ /\*\*
- ☐ /\*
- ☐ /\*=
- ☒ ///

## CORRECTION

**Question 38 ♣** Avec Git, je crée un nouveau fichier 'src/fic.cpp', quelle(s) commande(s) dois-je entrer pour que les autres développeurs puissent récupérer ce fichier ? (ne vous préoccupez pas de l'ordre, cochez toutes les commandes nécessaire).

- ☐ git commit
- ☐ git add
- ☒ git add src/fic.cpp
- ☒ git commit -m "ajout de fic" .
- ☐ git commit -a src/fic.cpp
- ☒ git push
- ☐ git a src/fic.cpp
- ☐ git commit fic.cpp

**Question 39 ♣** Cochez les balises Doxygen valides.

- ☐ \run pour décrire l'exécution du code.
- ☒ \brief pour donner une description courte.
- ☐ \save pour sauvegarder toutes les données dans un fichier d'historique.
- ☐ \ask pour poser une question à l'utilisateur.
- ☒ \param pour documenter un paramètre de fonction/méthode.
- ☒ \fn pour documenter une fonction.
- ☒ \struct \class pour documenter une structure/class C/C++.

**Question 40 ♣** 'core dump' signifie que le système a

- ☐ sauvegardé l'historique des versions de code dans un fichier 'core'.
- ☐ sauvegardé l'historique des instructions dans un fichier 'core'.
- ☒ sauvegardé tout l'état de la mémoire dans un fichier 'core'.

**Question 41 ♣** Un 'callback'

- ☒ est une fonction qui sera appelée par la boucle d'événement.
- ☒ est représenté par un pointeur de fonction.
- ☐ est une fonction qui s'appelle elle-même.

**Question 42 ♣** Cochez les méthodes de conceptions vues durant le cours :

- ☒ Méthode AGILE.
- ☒ Cycle en spiral.
- ☐ Méthode SOUPLE.
- ☐ Cycle en W.
- ☐ Cycle procédural.
- ☒ Cycle itératif.
- ☒ Cycle en V.
- ☒ Cycle en cascade.
- ☐ Cycle en itératif.

**Question 43 ♣** Doxygen

- ☒ utilise la grammaire du langage dans lequel est écrit le code source, ainsi que les commentaires.
- ☐ récupère et traite des actions de la souris.
- ☒ est un générateur de documentation capable de produire une documentation à partir du code source d'un programme.
- ☒ est sous licence libre.
- ☐ gère les versions du code d'un programme.

## CORRECTION

- Question 44 ♣** Cette commande 'g++ -ggdb -o projet main.o -L/usr/lib -lABC'
- ☒ '-lABC', indique qu'il faut inclure la bibliothèque (library) ABC à l'exécutable.
  - ☒ '-L/usr/lib' indique un chemin vers les bibliothèques (.a, .so ou .dll).
  - ☒ effectue la phase d'édition de lien.
  - ☐ effectue la phase de compilation.
  - ☒ crée un exécutable en assemblant les fichiers objets.
  - ☐ '-lABC' indique qu'il faut inclure le ABC.h à la compilation.
  - ☐ demande la production de fichiers objets (.o).
  - ☐ '-L/usr/lib' indique un chemin vers les fichiers d'entêtes (.h).