



+160/1/30+

<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input checked="" type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input checked="" type="checkbox"/>	0	<input type="checkbox"/>	0
<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	1	<input checked="" 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 checked="" 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 checked="" 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 checked="" 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 checked="" 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 : BROILLET
Prénom : Virgile

**LIFAPCDA Conception et développement d'applications : Examen (CC
mi-parcours) du 28 février 2023 (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.
Attention : les mauvaises réponses ou les non-réponses peuvent donner des points négatifs.

Question 1 Avec Git, la commande 'git log'

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

1/1

Question 2 Avec 'git commit' quelle option permet d'ajouter un commentaire ?

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

1/1

Question 3 ♣ La fonction 'SDL_RenderPresent' de SDL2

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

2/2

Question 4 ♣ Cochez les actions qui sont exécutées classiquement par une boucle d'événement dans une IHM

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

2/2



Question 5 Je veux supprimer un fichier de mon répertoire de travail et que git le prenne en compte (il restera dans l'historique des versions passées) :

- ☐ rm lefichier
- ☐ je ne peux pas simplement
- ☒ git rm lefichier
- ☐ git -remove lefichier

Question 6 ♣ Qt permet facilement

- ☒ de faire saisir à l'utilisateur du texte.
- ☒ de réagir à un événement clavier.
- ☒ d'ajouter des menus à une application.
- ☒ d'afficher des images.
- ☒ de jouer des sons.
- ☒ d'ajouter des bar d'icônes à une application.

Question 7 ♣ Dans le module Image, avec le code
`Image im1(256,256);`

- ☐ Le tableau de Pixel de im1 est stocké sur la pile.
- ☐ La variable im1 est stocké dans le tas.
- ☒ La variable im1 est stocké sur la pile.
- ☒ Le tableau de Pixel de im1 est stocké dans le tas.

Question 8 ♣ Un débogueur sert à

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

Question 9 ♣ Ce code

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

- ☐ provoque une erreur de compilation.
- ☒ peut s'exécuter sans erreurs apparentes.
- ☐ est un code correct.
- ☒ provoque une erreur avec Valgrind.
- ☒ peut provoquer un 'segmentation fault (core dump)'.

Question 10 ♣ Cette commande '`g++ -ggdb -o projet main.o truc.o -L/usr/lib -lABC`'

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



Question 11 ♣ Dans le module Image, avec le code
`Image* im2 = new Image(256,256);`

- ☒ Il faudra faire un delete im2
- ☒ Le tableau de Pixel de im2 est stocké dans le tas.
- ☐ Le pointeur im2 est stocké dans le tas.
- ☒ Ce qui est pointé par im2 est stocké dans le tas.
- ☐ Il faudra faire un delete [] im2;
- ☒ Le pointeur im2 est stocké sur la pile.

Question 12 ♣ Cochez les balises Doxygen valides.

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

Question 13 ♣ SDL permet facilement

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

Question 14 Le type C++ la déclaration d'un entier compris entre 0 et 255 est

- ☐ char
- ☐ int
- ☐ short
- ☒ unsigned char

Question 15 ♣ Mon programme ne fait pas ce qu'il devrait faire, mais ne plante pas. Quelle solution ai-je pour le corriger ?

- ☐ Corriger les erreurs de compilation.
- ☒ Comprendre ce qu'il fait réellement avec un débogueur.
- ☒ Comprendre ce qu'il fait réellement avec des affichages 'cout/printf'.
- ☒ Commenter certaines parties pour me concentrer sur une sous-partie du code.
- ☒ Le tester avec Valgrind.

Question 16 ♣ Cochez les règles d'écriture de code importantes.

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

Question 17 ♣ Un 'callback'

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



Question 18 ♣ Pour gérer toutes les notes d'une UE nous écrivons la classe suivante :

```
0: class LIFAP
1: {
2: public:
3:     LIFAP() : nbNotes(0), tailleAllouee(10), notes(nullptr) {}
4:     int sauver(const std::string& filename) const;
5:     void ajouterUneNote(int n) { CODE }
6: private:
7:     int *notes;
8:     int nbNotes, tailleAllouee;
9: };
```

- ☐ Ligne 3 : il manque une/des initialisation(s).
- ☒ Ligne 4 : le const à la fin est correcte.
- ☒ Ligne 7 : notes est un pointeur.
- ☒ ligne 5 : les instructions CODE feront une (re)allocation dynamique si le tableau est plein.
- ☒ Un destructeur serait utile dans cette classe.
- ☒ Ligne 3 : une ou plusieurs initialisations sont fausses.
- ☒ Ligne 7 : notes est un tableau.
- ☐ Ligne 4 : le 'const' à la fin signifie que la fonction renvoie un entier constant.
- ☐ Ligne 3 : les initialisations devraient être entre les accolades.

Question 19 ♣ Doxygen

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

Question 20 ♣ La directive ci-dessous

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

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

Question 21 ♣ Pour comprendre, inspecter et déboguer la gestion de la mémoire d'un programme je peux utiliser

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



Question 22 Avec Git, la copie de travail

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

Question 23 Dans le module Image, pour récupérer le Pixel original de coordonnée (x,y), quelle est la bonne manière d'écrire la fonction dans la classe Image?

- ☐ Pixel* getPix(int x, int y);
- ☒ Pixel& getPix(int x, int y);
- ☐ Pixel getPix(int x, int y);

Question 24 ♣ Ce code

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

- ☒ est source de confusion.
- ☐ provoque une erreur avec Valgrind.
- ☒ ne provoque pas d'erreur.
- ☐ provoque un 'segmentation fault (core dump)'.
- ☒ provoque une erreur avec un débogueur.

Question 25 'core dump' signifie que le système a

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

Question 26 Avec Git, le dépôt distant

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

Question 27 ♣ Avec SDL2, une SDL_Surface

- ☒ peut-être recopier directement à l'écran.
- ☒ peut être initialisée directement à partir d'un fichier.
- ☐ est une image spécifique au driver graphique stocké dans la mémoire de la carte graphique.
- ☒ est une image en mémoire centrale.

Question 28 Avec Git, la commande git pull

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

Question 29 Dans le module Image, cette année l'image représentait

- ☒ une maison
- ☐ un poisson
- ☐ une voiture
- ☒ un personnage vert (Yoda)



Question 30 ♣ Avec Git, quelle commande a besoin d'un accès réseau?

- ☐ git commit
☒ git push
☒ git pull

Question 31 ♣ Le cahier des charges

- ☒ définit exhaustivement les spécifications de l'application.
☐ propose des cycles AGILE de conception.
☒ indique les modalités d'exécution pour la réalisation avec les éléments/versions à livrer.
☐ définit le nombre de niveaux qu'aura le jeu.
☐ indique le nom des développeurs pour chaque tâche.
☒ est un contrat entre le client et le prestataire de service.

Question 32 ♣ Avec Git, le dépôt local

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

Question 33 Avec Git quelle est l'opération inverse de 'git add'?

- ☐ git reset HEAD
☒ git rm fichier
☐ rm

Question 34 ♣ La fonction 'SDL_RenderCopy' de SDL2

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

Question 35 ♣ La commande 'g++ -Wall -ggdb -c main.cpp -I/usr/include'

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

Question 36 Avec Git, la commande 'git commit'

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

Question 37 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 38 Dans le module Image, un unique Pixel occupe

- ☒ 3 x sizeof(int) octets
☒ 3 octets



Question 39 ♣ 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 ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois, etc.).
- ☒ La colonne de gauche du diagramme énumère toutes les tâches à effectuer.
- ☒ Un diagramme de Gantt permet de visualiser dans le temps les diverses tâches composant un projet.

Question 40 ♣ Doxygen analyse les commentaires commençant par

- ☒ /**
- ☒ ///
- ☐ /*=
- ☐ /*

Question 41 ♣ Ce code
`assert(a!=0);`

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

Question 42 ♣ Avec SDL2, une `SDL_Texture`

- ☒ peut-être recopier directement à l'écran.
- ☒ représente une image spécifique au driver graphique stocké dans la mémoire de la carte graphique.
- ☐ représente une image en mémoire centrale.
- ☒ peut être initialisée directement à partir d'un fichier.

Question 43 ♣ Segmentation fault (core dumped) signifie

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

Question 44 ♣ La balise `\brief` sert à

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

Question 45 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 46 ♣ Même si un programme fonctionne, est-ce que Valgrind peut servir quand même ?

- ☒ Oui, à détecter des fuites mémoires.
- ☐ Non
- ☒ Oui, à examiner les performances.
- ☒ Oui, à détecter des erreurs subtiles de gestion de la mémoire.
- ☐ Oui, à détecter des erreurs de compilation.

Question 47 Ce code

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

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

Question 48 Une bonne manière de passer en donnée un argument de type Camion (avec `sizeof(Camion)>8`) à une fonction est

- ☐ `void bonneFonction(Camion c);`
- ☐ `void bonneFonction(Camion& c);`
- ☒ `void bonneFonction(const Camion& c);`

Question 49 ♣ Cochez les affirmations appartenant à la méthode de conception AGILE

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

Question 50 ♣ Avec Git, je crée un nouveau fichier 'src/fic.cpp', quelle(s) commande(s) doit-on entrer pour que les autres développeurs puissent récupérer ce fichier ? (ne vous préoccupez pas de l'ordre, cochez toutes les commandes nécessaires).

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



Question 51 ♣ Cochez les méthodes de conceptions vues durant le cours :

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

Question 52 ♣ Dans un Makefile, il y a des règles

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

0.2/1

2/2



+160/10/21+