

# Algèbre de Boole - Quiz

<https://github.com/heig-vd-progim-course/heig-vd-progim2-course>

Visualiser le contenu complet sur GitHub [à cette adresse](#).

V. Guidoux, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

## Plus de détails sur GitHub

*Cette présentation est un quiz pour tester ses connaissances sur le chapitre en cours. Pour plus de détails, consultez le [contenu complet sur GitHub](#).*

# Question 1 - Donnée

## Complétion : Déclaration de variable booléenne

Quel code déclare correctement une variable booléenne `isActive` initialisée à vrai ?

- **A.** `boolean isActive = 1;`
- **B.** `boolean isActive = true;`
- **C.** `bool isActive = true;`
- **D.** `Boolean isActive = "true";`

# Question 1 - Réponse

**Réponse correcte : B**

```
boolean isActive = true;
```

**Explications :**

- **A** : `1` n'est pas une valeur booléenne valide en Java
- **C** : le type est `boolean`, pas `bool`
- **D** : `"true"` est une chaîne de caractères, pas une valeur booléenne

## Question 2 - Donnée

### Complétion : Opérateur logique manquant

Complétez le code pour vérifier qu'une personne a accès SI elle a un badge **ET** un code valide :

```
boolean hasBadge = true;  
boolean hasValidCode = true;  
boolean hasAccess = hasBadge ___ hasValidCode;
```

**A.** || **B.** && **C.** ! **D.** ^

## Question 2 - Réponse

Réponse correcte : B

```
boolean hasAccess = hasBadge && hasValidCode;
```

Explications :

- **&&** (AND) : les **deux** conditions doivent être vraies
- **||** (OR) : au moins une condition doit être vraie
- **!** (NOT) : inverse la valeur
- **^** (XOR) : vrai si les valeurs sont différentes

## Question 3 - Donnée

### Complétion : Application de la loi de De Morgan

Quelle expression est équivalente à `!(isOffline || hasError)` ?

- **A.** `isOffline || hasError`
- **B.** `!isOffline || !hasError`
- **C.** `!isOffline && !hasError`
- **D.** `isOffline && hasError`

## Question 3 - Réponse

### Réponse correcte : C

`!(isOffline || hasError)` équivaut à `!isOffline && !hasError`

### Loi de De Morgan :

- $!(A \ || \ B) = !A \ \&\& \ !B$
- On retire la négation globale
- On inverse l'opérateur ( `||` → `&&` )
- On ajoute une négation à chaque terme

## Question 4 - Donnée

### Prédiction : Évaluation

Que vaut `result` après l'exécution de ce code ?

```
boolean a = false;  
boolean b = true;  
boolean result = a && b;
```

- **A.** `true`
- **B.** `false`
- **C.** Le code ne compile pas
- **D.** Une exception est levée

# Question 4 - Réponse

## Réponse correcte : B

```
boolean result = a && b; // result vaut false
```

## Explications :

- L'opérateur `&&` évalue à `true` uniquement si les deux conditions sont vraies
- Ici, `a` est `false`, donc le résultat est forcément `false`
- Grâce au court-circuit, `b` n'est même pas évalué (optimisation)

## Question 5 - Donnée

### Prédiction : Expression avec plusieurs opérateurs

Que vaut `canAccess` après l'exécution de ce code ?

```
boolean isAdmin = false;  
boolean hasPermission = true;  
boolean isVerified = true;  
boolean canAccess = isAdmin || (hasPermission && isVerified);
```

**A.** `true` **B.** `false` **C.** Erreur de compilation **D.** `null`

## Question 5 - Réponse

**Réponse correcte : A** `canAccess` vaut `true`

```
boolean isAdmin = false;  
boolean hasPermission = true;  
boolean isVerified = true;  
boolean canAccess = isAdmin || (hasPermission && isVerified); // true
```

**Évaluation étape par étape :**

- `hasPermission && isVerified` → `true && true` = `true`
- `isAdmin || true` → `false || true` = `true`

## Question 6 - Donnée

### Prédiction : Opérateur XOR

Que vaut `result` après l'exécution de ce code ?

```
boolean solarPower = true;  
boolean gridPower = true;  
boolean result = solarPower ^ gridPower;
```

- **A.** `true`
- **B.** `false`
- **C.** Erreur de compilation
- **D.** Dépend de la JVM utilisée

## Question 6 - Réponse

**Réponse correcte : B** result vaut false

```
boolean solarPower = true;  
boolean gridPower = true;  
boolean result = solarPower ^ gridPower;
```

### Explications :

- L'opérateur `^` (XOR) retourne `true` si les valeurs sont **différentes**

## Question 7 - Donnée

### Comparaison : Conditions redondantes

Quelle version est préférable pour vérifier si un nombre est positif ?

- **A.** `if (number > 0 == true) { ... }`
- **B.** `if ((number > 0) == true) { ... }`
- **C.** `if (number > 0) { ... }`
- **D.** Les trois sont équivalentes et acceptables

## Question 7 - Réponse

**Réponse correcte : C** `if (number > 0) { ... }`

### Explications :

- `number > 0` retourne déjà un `boolean`
- Comparer un booléen avec `== true` est redondant
- Code plus lisible et idiomatique
- Principe KISS (Keep It Simple and Stupid)

**À éviter :** comparaisons redondantes comme `== true` ou `== false`

## Question 8 - Donnée

### Comparaison : Négations multiples

Quelle version est la plus lisible et maintenable ?

```
// Version A
if (!(!isConnected || !hasData)) { ... }

// Version B
if (isConnected && hasData) { ... }
```

**A.** Version A **B.** Version B **C.** Équivalentes **D.** Aucune

# Question 8 - Réponse

## Réponse correcte : B

```
if (isConnected && hasData) { ... }
```

## Explications :

- Les deux versions sont **logiquement équivalentes**
- Version B est plus simple et directe
- Éviter les négations multiples améliore la lisibilité
- Application des lois de De Morgan pour simplifier

## Question 9 - Donnée

### Modification : Simplification avec De Morgan

Comment simplifier cette condition ?

```
if (!(temperature < 15 || humidity > 80)) {  
    startSystem();  
}
```

- A `if (temperature >= 15 || humidity <= 80)`
- B `if (temperature >= 15 && humidity <= 80)`
- C `if (!(temperature >= 15 && humidity <= 80))`
- D `if (temperature < 15 && humidity > 80)`

# Question 9 - Réponse

**Réponse correcte : B**

```
if (temperature >= 15 && humidity <= 80) {  
  ...  
}
```

**Application de la loi de De Morgan :**

1.  $\!(A \ || \ B) = \!A \ \&\& \ \!B$
2.  $\!(\text{temperature} < 15) \rightarrow \text{temperature} \geq 15$
3.  $\!(\text{humidity} > 80) \rightarrow \text{humidity} \leq 80$

# Questions

Est-ce que vous avez des questions sur le quiz ou le contenu ?

# Sources

- [Illustration principale](#) par [Richard Jacobs](#) sur [Unsplash](#)