**Ερωτήσεις Πολλαπλής Επιλογής στους Τελεστές της C++**

**Αριθμητικοί Τελεστές & Προτεραιότητα**

Ποιο θα είναι το αποτέλεσμα της έκφρασης int result = 5 + 3 \* 2; στη C++;

A) 16

B) 11

C) 8

D) Σφάλμα μεταγλώττισης

Τι θα εμφανίσει ο παρακάτω κώδικας C++; cout << 10 / 4;

A) 2.5

B) 3

C) 2

D) Σφάλμα

**Ακέραια & Πραγματική Διαίρεση**

Ποιο θα είναι το αποτέλεσμα της έκφρασης (float)9 / 2;

A) 4

B) 5

C) 4.5

D) Σφάλμα χρόνου εκτέλεσης

Τι θα συμβεί κατά την εκτέλεση της εντολής cout << 5 / 0; σε ένα τυπικό περιβάλλον C++;

A) Θα τυπώσει 0

B) Θα τυπώσει 'inf' (άπειρο)

C) Μη καθορισμένη συμπεριφορά, πιθανότατα σφάλμα χρόνου εκτέλεσης (runtime error).

D) Θα προκαλέσει σφάλμα μεταγλώττισης (compilation error).

**Τελεστής Υπολοίπου (Modulo)**

Ποιο είναι το αποτέλεσμα της έκφρασης 7 % 3 στη C++;

A) 1

B) 2

C) 2.33

D) 0

Τι θα επιστρέψει η έκφραση -5 % 2 στη C++;

A) 1

B) -1

C) 2.5

D) -2.5

**Τελεστές Αυξομείωσης**

Δίνεται ο κώδικας: int i = 5; cout << i++;. Τι θα τυπωθεί στην οθόνη;

A) 6

B) 5

C) 4

D) Σφάλμα

Δίνεται ο κώδικας: int i = 0; int a = ++i;. Ποιες θα είναι οι τελικές τιμές των a και i;

A) a = 0, i = 1

B) a = 1, i = 0

C) a = 0, i = 0

D) a = 1, i = 1

**Σύγκριση Πραγματικών Αριθμών**

Γιατί η απευθείας σύγκριση δύο αριθμών float με τον τελεστή == θεωρείται κακή πρακτική στη C++;

A) Επειδή είναι πιο αργή από άλλες μεθόδους.

B) Επειδή μπορεί να οδηγήσει σε σφάλμα μεταγλώττισης.

C) Λόγω πιθανών σφαλμάτων στρογγυλοποίησης στην αναπαράσταση των δεκαδικών αριθμών.

D) Επειδή ο τελεστής == λειτουργεί μόνο με ακέραιους.

Ποια έκφραση είναι η καταλληλότερη για να ελέγξει αν δύο μεταβλητές double, d1 και d2, είναι "ουσιαστικά ίσες";

A) d1 == d2

B) fabs(d1 - d2) < 0.000001

C) d1 - d2 == 0

D) (int)d1 == (int)d2

**Συνδυαστική Ερώτηση**

Ποιος κώδικας θα αποθηκεύσει τον αριθμό 2 στη μεταβλητή c;

A) int a = 5, b = 2; int c = ++a % b;

B) int a = 8, b = 3; int c = a / b;

C) int a = 1, b = 2; int c = a + b--;

D) int a = 9, b = 4; int c = b - (a % b);