1. What will the following code display?  
   enum { POODLE, BOXER, TERRIER };  
   cout << POODLE << " " << BOXER << " " << TERRIER << endl;

**0 1 2**

1. Look at the following declaration.  
   enum Person { BILL, JOHN, CLAIRE, BOB };  
   Person p;  
   Indicate whether each of the following statements or expressions is valid or invalid.  
   **A) p = BOB;**  
   B) p++;  
   **C) BILL > BOB**  
   D) p = 0;  
   **E) int x = BILL;**  
   **F) p = static\_cast<Person>(3);  
   G) cout << CLAIRE << endl;**
2. Look at the following statement.  
   enum Color { RED, ORANGE, GREEN, BLUE };  
   A)What is the name of the data type declared by this statement?

**Color**

B)What are the enumerators for this type?

**RED, ORANGE, GREEN, BLUE**

C)Write a statement that defines a variable of this type and initializes it with a valid value.

**Color example = RED;**

1. A pet store sells dogs, cats, birds, and hamsters. Write a declaration for an enumerated data type that can represent the types of pets the store sells.

**enum PetType { DOG, CAT, BIRD, HAMSTER };**

1. Mark the following statements as **true** or false.

**a. The following is a valid C++ enumeration type:**

**enum romanNumerals {I, V, X, L, C, D, M};**

b. Given the declaration:

enum cars {FORD, GM, TOYOTA, HONDA};

cars domesticCars = FORD;  
the statement:

domesticCars = domesticCars + 1;  
sets the value of domesticCars to GM.

**c. A function can return a value of an enumeration type.**

d. You can input the value of an enumeration type directly from a standard input device.

e. The only arithmetic operations allowed on the enumeration type are increment and decrement.

**f. The values in the domain of an enumeration type are called enumerators.**

**g. The following are legal C++ statements in the same block of a C++ program:  
enum mathStudent {BILL, JOHN, LISA, RON, CINDY, SHELLY};  
enum historyStudent {AMANDA, BOB, JACK, TOM, SUSAN};**

**h. The following statement creates an anonymous type:**

**enum {A, B, C, D, F} studentGrade;**

1. Write C++ statements that do the following:  
   a. Define an enum type, bookType, with the values MATH, CSC, ENGLISH, HISTORY, PHYSICS, and PHILOSOPHY.  
   b. Declare a variable book of type bookType.  
   c. Assign MATH to the variable book.  
   d. Advance book to the next value in the list.  
   e. Output the value of the variable book.

**enum bookType { MATH, CSC, ENGLISH, HISTORY, PHYSICS, PHILOSOPHY };**

**bookType book;**

**book = MATH;**

**book = static\_cast<bookType>(book + 1);**

**cout << book;**

1. Given:  
   enum currencyType {DOLLAR, POUND, FRANK, LIRA, MARK};  
   currencyType currency;  
   which of the following statements are valid?  
   **a. currency = DOLLAR;**  
   b. cin >> currency;  
   **c. currency = static\_cast<currencyType>(currency + 1);**  
   d. for (currency = DOLLAR; currency <= MARK; currency++)

cout << "\*";

1. What is the output of the following program?

#include <iostream>

using namespace std;

enum color\_type {red, orange, yellow, green, blue, violet};

main()

{

color\_type shirt, pants;

shirt = red;

pants = blue;

cout << shirt << " " << pants << endl;

return 0;

}

1. **4**
2. Given the following declaration and output statement, assume that this has been embedded in a correct program and is run. What is the output?  
   *enum* Direction { N, S, E, W };  
   //...  
   cout << W << " " << E << " " << S << " " << N << endl;

**3 2 1 0**

1. Given the following declaration and output statement, assume that this has been embedded in a correct program and is run. What is the output?  
   *enum* Direction { N = 5, S = 7, E = 1, W };  
   // ...  
   cout << W << " " << E << " " << S << " " << N << endl;

**2 1 7 5**