1. The following program segment does not produce the correct results. Why? How can it be fixed?

```cpp
char   quantity;
double cost;
cin >> quantity >> cost;
cout << "Total is " << quantity * cost << endl;
```

2. Acme Department Store has new product information stored in a file whose lines consist of product id, product type and cost. The product id is a **long** integer while the cost is a **double**. The product type is a letter code indicating the category of merchandise, such as A (automotive), C (clothing), S (sports) and so on. A typical line looks like this with one blank on either side of the product type letter.

```
23455 A 4.99
```

However, since these are new items, sometimes the type of product has not yet been determined and that field is blank in that line. The programmer wrote the following input function. It does not work. Why? What can be done to fix it up so that it properly inputs the data whether or not the product type is temporarily blank?

```cpp
istream& GetData (istream& infile, long& id, char& type, double& cost) {
    infile >> id >> type >> cost;
    return infile;
}
```

3. Another Acme programmer attempted to fix the program in Problem 2 above by coding the following function. It does not work properly either. Why? How could it be fixed to work correctly?

```cpp
istream& GetData (istream& infile, long& id, char& type, double& cost) {
    infile >> id;
    infile.get (type);
    infile >> cost;
    return infile;
}
```
4. What is wrong with the following Do Case coding? How can it be fixed up so that it would work?

```cpp
double month;
switch (month) {
case 1:
case 2:
case 12:
    // winter costs are 25% higher
    double sum;
    sum = qty * cost * 1.25;
    break;
default:
    double sum;
    sum = qty * cost;
    break;
}
cout << sum;
```

5. The programmer goofed while coding this Do Case to calculate the shift bonus for the employee payroll. What is wrong and how can it be fixed?

```cpp
char shift;
switch (shift) {
    case '1':
        pay = hours * rate;
        case '2':
        pay = hours * rate * 1.05;
        case '3':
        pay = hours * rate * 1.12;
}
cout << pay;
```

6. The programmer wanted to setup an enumerated data type to handle the employee’s shift. However, the following coding fails. Why? How can it be repaired?

```cpp
Enum ShiftType = First, Second, and Third;
```

7. A programmer setup the following `enum` to handle the product types.

```cpp
enum ProductTypes {Games, Auto, Clothing, Appliances};
```
In the input a set of data function, the user is instructed to enter a letter for the product type: G, A, C or A. What is the design flaw and why does not the following input coding work? How can it be repaired?

```cpp
ProductTypes prodType;
```
8. In Problem 7 above, the programmer got frustrated and then did the following which does not compile. Why? Can this coding be repaired?

```cpp
ProductTypes prodType;
char c;
infile >> c;
if (c == 'A')
    prodType = 1;
if (c == 'G')
    prodType = 0;
```