

OOP Lab Final Exam (Fall 2022 - BSCS)

Allowed Time = 90 minutes

Version: X1

Instructions for Invigilators:

1. Students will have total 90 minutes to finish the whole exam.
2. It is up to the students to manage their time.

Instructions for Students:

1. There is only 1 question in the exam.
2. Submit **.h** and **.cpp** files only.
3. Late submissions will **NOT** be considered
4. Create as many classes and functions as required. Remember, one function for one functionality.
5. Plagiarism will not be tolerated in any case.
6. Use meaningful variable names, and take care of naming conventions and indentation.
7. The format of the name of your submission should be
 - *RegisterionNumber_Name_Course_Section.*
 - For eg: **L1F13BSCS2124_MohsinAbbas_OOP_SEC_D**

Time allowed = 90 minutes;

Question: Write the C++ program to create a polymorphic class hierarchy based on the code and the corresponding output below. The program should ensure there is no memory leakage or shallow copy.

Code

```
double Product::totalPrice = 0.0;

int main()
{
    // Create some products and add them to the array
    const int MAX_PRODUCTS = 4;
    Product *products[MAX_PRODUCTS]; //Product is Abstract class
    products[0] = new Grocery("Apple", 0.5);
    products[1] = new Dairy("Cheese", 2.0);
    products[2] = new Meat("Chicken", 3.0);
    products[3] = new Electronics("TV", 500.0);

    // Display the total price of all products
    cout << "Price of all products: $"
         << Product::getTotalPrice() << endl;

    // Print the information of all the products in the array
    for (int i = 0; i < MAX_PRODUCTS; i++)
        products[i]->describe();

    // Delete the products
    for (int i = 0; i < MAX_PRODUCTS; i++)
        delete products[i];

    return 0;
}
```

Output

```
Price of all products: $505.5
Apple (grocery item) - price: $0.5
Cheese (dairy item) - price: $2
Chicken (meat item) - price: $3
TV (electronics item) - price: $500
~Grocery()
~Product()
~Dairy()
~Product()
~Meat()
~Product()
~Electronics()
~Product()
```