CC-11C-L: Object Oriented Programming Lab
BSSE F23 – Semester: Spring 2024
Lab 10

Roll-No

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Issue Date: 29-Apr-2024
Marks: 16
Time: 140 minutes

Objective:

- It will help you understand the benefits that we get through inheritance relationship.

Challenge – 1: (16)

Define a class named **Payment** that contains an instance variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named paymentDetails that outputs an English sentence to describe the amount of the payment.
Next, define a class named **CashPayment** that is derived from Payment. This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s).
Define a class named **CreditCardPayment** that is derived from Payment. This class should contain instance variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout.
Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

Sample Run

Sample Code	Sample Output

CashPayment cp1(100), cp2(500);
CreditCardPayment
    ccp1(200, "Kamran", Date(15, 4, 2024), "2398573957"),
    ccp2(50, "Aslam", Date(1, 2, 2020), "123456789");	

cp1.paymentDetails();	The payment of cash: $100
cp2.paymentDetails();	The payment of cash: $500
ccp1.paymentDetails();	The payment of $200 through the card 2398573957, and expire date 15-04-2024, and the owner’s name: Kamran.
ccp2.paymentDetails();	The payment of $50 through the card 123456789, and expire date 01-02-2020, and the owner’s name: Aslam.

Challenge – 2: (16)

Define a class named **Document** that contains a member variable of type String named text that stores any textual content for the document. Provide appropriate constructor for this class.
Next, define a class for **Email** that is derived from Document and that includes member variables for the sender, recipient, and title of an e-mail message. Implement constructor and appropriate accessor and mutator functions. The body of the e-mail message should be stored in the inherited variable text.
Similarly, define a class for **File** that is derived from Document and that includes a member variable for the pathName and fileName. Implement appropriate accessor and mutator functions for them as well.
Implement a method named as toString in the Document class that returns a text field in Document.
Redefine toString method in class Email that concatenate all the fields/attributes appropriately/meaningfully (see sample sun). In the same way, redefine toString for class File as well.
Finally, create several sample objects of type Email and File in your main function. Test your objects by passing them to the following subroutine SearchRoutines::containsKeyword, which will return true if the object contains the specified keyword in the text property.

```
class SearchRoutines
{
public:
    static bool containsKeyword(const Document & docObject, const String & keyWord);
};
```

Sample Run

Line #	Sample Code
1	int main()
2	{
3	Email emailObj("bsef23m001@pucit.edu.pk", "bsef23m002@pucit.edu.pk", "My Best Friend", "My best friend’s name is Laiba, and we’ve known each other since nursery. We live in the same colony and share countless memories from our early days. Laiba is a true leader in every sense. She excels in her studies, always impressing our teachers with her dedication. We often share our lunch and love playing carom at home. Music and cartoons are our favourite pastimes, bringing us even closer. Laiba is always punctual and values good manners, setting a great example for all. We help each other with homework, especially if one of us misses school. Together, we eagerly await family trips during the vacations, making memories that last a lifetime.");
4	File fileObj("c:\\abc\\work\\file.txt", "Test.txt", "//This is some sample text in a file containing C++ code.\nbool ContainsKeyword(const Document & docObject, const String & keyword)\n{\n\treturn true;\n}");
5	if (SearchRoutines::containsKeyword(emailObj, "manners");
6	cout << "The email contains the keyword 'manners'." << '\n';
7	else
8	cout << "The email does NOT contain the keyword 'manners'." << '\n';
9	if (SearchRoutines::containsKeyword(fileObj, "manners"))
10	cout << "The file contains the keyword 'manners'." << '\n';
11	else
12	cout << "The file does NOT contain the keyword 'manners'." << '\n';
13	cout << emailObj.toString();
14	cout << '\n' << fileObj.toString();
15	return 0;
16	}

Line #	Sample Output
1	
2	
3	
4	
5	The email contains the keyword 'manners'.
6	
7	
8	
9	The file does NOT contain the keyword 'manners'.
10	
11	
12	
13	Sender: bsef23m001@pucit.edu.pk Recipient: bsef23m002@pucit.edu.pk Title: My Best Friend Message: My best friend’s name is Laiba, and we’ve known each other since nursery. We live in the same colony and share countless memories from our early days. Laiba is a true leader in every sense. She excels in her studies, always impressing our teachers with her dedication. We often share our lunch and love playing carom at home. Music and cartoons are our favorite pastimes, bringing us even closer. Laiba is always punctual and values good manners, setting a great example for all. We help each other with homework, especially if one of us misses school. Together, we eagerly await family trips during the vacations, making memories that last a lifetime.

14	File Path: c:\abc\work\file.txt
	File Name: Test.txt
	Contents: //This is some sample text in a file containing C++ code.
	bool ContainsKeyword(const Document & docObject, const String & keyword)
	{
	    return true;
	}

15	
16	