**Lab 10**

**Object Oriented Programming Lab**

**Common Solution**

**13 Marks**

**Challenge-1: (5)**

**Payment.h**

#ifndef PAYMENT\_H

#define PAYMENT\_H

#include<iostream>

using namespace std;

class Payment

{

double amount;

public:

Payment(double);

double getAmount() const;

void setAmount(double);

void paymentDetails() const;

};

#endif

**Payment.cpp**

#include"Payment.h"

Payment::Payment(double amt) : amount(amt)

{

}

double Payment::getAmount() const

{

return amount;

}

void Payment::setAmount(double amt)

{

amount = amt;

}

void Payment::paymentDetails() const ----(0.5)

{

cout << "Payment amount: $" << amount << endl;

}

**CashPayment.h**

#ifndef CASH\_PAYMENT\_H

#define CASH\_PAYMENT\_H

#include"Payment.h"

class CashPayment : public Payment

{

public:

CashPayment(double);

void paymentDetails() const;

};

#endif

**CashPayment.cpp**

#include"CashPayment.h"

CashPayment::CashPayment(double amt) : Payment(amt) ----(0.5)

{

}

void CashPayment::paymentDetails() const ----(1)

{

cout << "The payment of cash: $" << getAmount() << endl;

}

**CreditCardPayment.h**

#ifndef CREDIT\_CARD\_PAYMENT\_H

#define CREDIT\_CARD\_PAYMENT\_H

#include"Payment.h"

#include"BigNumber.h"

#include"Date.h"

class CreditCardPayment : public Payment

{

BigNumber cardNumber;

Date expirationDate;

String ownerName;

public:

CreditCardPayment(double, const String&, const Date&, const BigNumber&);

void paymentDetails() const;

};

#endif

**CreditCardPayment.cpp**

#include"CreditCardPayment.h"

CreditCardPayment::CreditCardPayment(double amt, const String& owner, const Date& expDate, const BigNumber& cardNum) : Payment(amt), cardNumber(cardNum), expirationDate(expDate), ownerName(owner) ----(1)

{

}

void CreditCardPayment::paymentDetails() const ----(2)

{

cout << "The payment of $" << getAmount() << " through the card ";

cardNumber.print();

cout << ", and expire date ";

expirationDate.printFormat1();

cout << ", and the owner's name: " << ownerName << "." << endl;

}

**Challenge-2: (7.5)**

**Document.h**

ifndef DOCUMENT\_H

#define DOCUMENT\_H

#include"String.h"

class Document

{

String text;

public:

Document(const String&);

String toString() const;

};

#endif

**Document.cpp**

#include"Document.h"

Document::Document(const String& t) :text(t)

{

}

String Document::toString() const ----(0.5)

{

return text;

}

**Email.h**

#ifndef EMAIL\_H

#define EMAIL\_H

#include"Document.h"

class Email :public Document

{

String sender;

String recipient;

String title;

public:

Email(const String&, const String&, const String&, const String&);

void setSender(const String&);

void setRecipient(const String&);

void setTitle(const String&);

String getSender() const;

String getRecipient() const;

String getTitle() const;

String toString() const;

};

#endif

**Email.cpp**

#include"Email.h"

Email::Email(const String& s, const String& r, const String& t, const String& txt):Document(txt),sender(s),recipient(r),title(t)

----(0.5)

{

}

void Email::setSender(const String& s)

{

sender = s;

}

void Email::setRecipient(const String& r)

{

recipient = r;

}

void Email::setTitle(const String& t)

{

title = t;

}

String Email::getSender() const

{

return sender;

}

String Email::getRecipient() const

{

return recipient;

}

String Email::getTitle() const

{

return title;

}

String Email::toString() const ----(2)

{

String txt = "Sender: " + sender + "\nRecipient: " + recipient + "\nTitle: " + title + "\nMessage: ";

txt += Document::toString();

return txt;

}

**File.h**

#ifndef FILE\_H

#define FILE\_H

#include"Document.h"

class File:public Document

{

String pathName;

String fileName;

public:

File(const String&, const String&, const String&);

void setPathName(const String&);

void setFileName(const String&);

String getPathName() const;

String getFileName() const;

String toString() const;

};

#endif

**File.cpp**

#include"File.h"

File::File(const String& p, const String& f, const String& t):Document(t),pathName(p),fileName(f) ----(0.5)

{

}

void File::setPathName(const String& p)

{

pathName = p;

}

void File::setFileName(const String& f)

{

fileName = f;

}

String File::getPathName() const

{

return pathName;

}

String File::getFileName() const

{

return fileName;

}

String File::toString() const ----(2)

{

String txt = "File Path: " + pathName + "\nFile Name: " + fileName + "\nContents: ";

txt += Document::toString();

return txt;

}

**SearchRoutines.h**

#ifndef SEARCH\_ROUTINES\_H

#define SEARCH\_ROUTINES\_H

#include"Document.h"

class SearchRoutines

{

public:

static bool containsKeyword(const Document&, const String&);

};

#endif

**SearchRoutines.cpp**

#include"SearchRoutines.h"

bool SearchRoutines::containsKeyword(const Document& docObject, const String& keyWord) ----(2)

{

return (docObject.toString().find(keyWord) == -1 ? false : true);

}

**Quick Revision:**

**Challenge-1: (5)**

void Payment::paymentDetails() const ----(0.5)

CashPayment::CashPayment() ----(0.5)

void CashPayment::paymentDetails() const ----(1)

Print exactly like sample run.

CreditCardPayment::CreditCardPayment() ----(1)

If not used BigNumber as cardNumber 🡪 0

void CreditCardPayment::paymentDetails() const ----(2)

if not printed like sample run 🡪 -1.5

**Challenge-2: (8)**

String Document::toString() const ----(0.5)

Email::Email() ----(0.5)

String Email::toString() const ----(2)

1. Atomicity, for not using String member functions for concatenation 🡪 -1
2. Not According to sample run 🡪 -1

File::File() ----(0.5)

String File::toString() const ----(2)

1. Atomicity, for not using String member functions for concatenation 🡪 -1
2. Not According to sample run 🡪 -1

bool SearchRoutines::containsKeyword() ----(2)

Atomicity to find 🡪 -1

For testing containsKeyword() function in main ---(0.5)

**Penalty Matrix:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Penalty List | Labs | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 3 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Indentation putting { Infront of loop header, in do while, putting while with closing } | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Meaningful Variable Names |  | -2 | -2 | -2 | -2 | -2 | -2 | -2 |  | -2 |  |  |  |  |  |  |
| Camel Case Notation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Atomicity |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Syntax error | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Linker error | 0 | 0 | 0 | 0 | 0 | 0 | - | - |  | - |  |  |  |  |  |  |
| Wrong function prototypes | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Class interface or additional members |  |  | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Use of library function/class without permission | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Continue statement | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| cin/cout where it isn’t needed | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Multi-filing |  |  | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  |  |
| Wrong #ifndef or name of header file |  |  | -2 | -2 | -2 | -2 | -2 | -2 |  | -2 |  |  |  |  |  |  |
| Global functions |  |  | -3 | -3 | -3 | -3 | -3 | -3 |  | -3 |  |  |  |  |  |  |
| Multiple classes in one header file |  |  | -3 | -3 | -3 | -3 | -50% | -50% |  | -50% |  |  |  |  |  |  |