# 

# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Term # \_\_\_\_

**Homework 11 - On Lecture 14 – Inheritance**

**(100 points) Hours:**

**The homework is to be turned in as a *PAPER AND PENCIL i.e., HANDWRITTEN ANSWER ONLY!(with your terminal #!)* in the first ten minutes of the due date class.**

**Also an implementation in Visual Studio is ALSO required, thus you are to submit the ZIPPED project to BB and download it in 232 PGH the first 10 minutes of class. Hardcopy with screenshots of the running program and the SOURCE CODE are also needed.**

**TURNING IN THE HOMEWORK INSTRUCTIONS will be PENALTY OF -10 points.**

**I UNDERSTAND THAT TURNING ANOTHER’s WORK IN is CHEATING.**

**I UNDERSTAND THAT ANY KIND OF DISSEMINATION of this WORK is CHEATING.**

**I CERTIFY THAT THE HOMEWORKs SOLUTIONs ARE MY OWN WORK!**

**?**

**X**

**V**

**SIGNATURE:**

**TA check, is Homework11.doc & Homework11.zip**

**in BB?**

**HOMEWORK CHECKLIST (YOU MUST GRADE YOURSELF!):**

1. **DID TURN IN HOMEWORK INSTRUCTIONS? \* -10 point**
2. **1.? 40 points**
3. **2.? H & E (attach to BB) – WORD ONLY 60 points**
4. **Homework11 C++.zip NOT submitted to BB? \* -50 points**
5. **Homework11 C++.zip NOT running in class? \* -50 points**

**\* If NOT, do not enter anything in the box!**

TA **grade or check**

**PLEASE ENTER YOUR GRADE IN THIS BOX:**

**By Hand?**

**Screenshot?**

# I understand that if the .zip file is NOT in BB and I did not check the BOX, I will get a ZERO for the Homework!

**1.** (40 pts)(**BY HAND**):

a. (8 pts) **Is** the following program legal ( assuming appropriate # include and using directives are added)?

void showEmployeeData( const Employee object);

int main( )

{

HourlyEmployee joe(" Mighty Joe", " 123- 45- 6789", 20.50, 40); SalariedEmployee boss(" Mr. Big Shot", " 987- 65- 4321", 10500.50); showEmployeeData( joe);

showEmployeeData( boss);

return 0;

}

void showEmployeeData( const Employee object)

{

cout << " Name: " << object. getName( ) << endl;

cout << " Social Security Number: " << object. getSsn( ) << endl;

}

**ANSWER:**

b. (8 pts The class SalariedEmployee inherits both of the functions getName and printCheck ( among other things) from the base class Employee, yet only the function declaration for the function printCheck is given in the definition of the class SalariedEmployee. Why isn’t the function declaration for the function getName given in the definition of SalariedEmployee?

**ANSWER:**

c. (8 pts)Give a definition for a class TitledEmployee that is a derived class of the base class SalariedEmployee given in Display 14.4. The class TitledEmployee has one additional member variable of type string called title. It also has two additional member functions: getTitle, which takes no arguments and returns a string, and setTitle, which is a void function that takes one argument of type string. It also redefines the member function setName. You do not need to give any implementations, just the class definition. However, do give all needed # include directives and all using namespace directives. Place the class TitledEmployee in the namespace SavitchEmployees.

**ANSWER:**

d. (8 pts)Give the definitions of the constructors for the class TitledEmployee that you gave as the answer to Self- Test Exercise 5. Also, give the redefinition of the member function setName. The function setName should insert the title into the name. Do not bother with # include directives or namespace details.

**ANSWER:**

e. (8 pts)You know that an overloaded assignment operator and a copy constructor are not inherited. Does this mean that if you do not define an overloaded assignment operator or a copy constructor for a derived class, then that derived class will have no assignment operator and no copy constructor?

**ANSWER:**

**2.** (60 pts) **UML Class Diagram** (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**).

Document

Define a Class named Document that contains a **private** member variable of type string named text that stores any textual content for the document. Create a function named getText that returns the text field, a way to set this value, and an **overloaded** assignment operator. Next, define a Class for Email that is derived from Document and that includes **private** member variables for the sender, recipient, and title of an e-mail message. Implement appropriate accessor and mutator functions. The body of the e-mail message should be stored in the **inherited** variable text. Also **overload** the assignment operator for this Class. Similarly, define a Class for File that is derived from Document and that includes a **private** member variable for the pathname. Implement appropriate accessor and mutator functions for the pathname and **overload** the assignment operator. Finally, create several sample objects of type Email and File in your main function. Test your objects by passing them to the following subroutine, which will return true if the object contains the specified keyword in the text property.

File: 5: setPathName (string):void

File: 4: getPathName ():string

Email: 7-9: getters():string

Email: 4-6: setters(string):void

Document: 2: setText(string):

File: 2: =(File&): operator

File: 1: pathname: string

File

Document: 3: =(Document&): operator

Email: 2: =(Email&): operator

Document: 1: getText(): string

Document: 1: text: string

Email: 3:title: string

Email: 2: recipient: string

Email: 1: sender: string

Email

bool containsKeyword( const Document& docObject, string keyword)

{

if ( docObject.getText().find( keyword) != string:: npos)

return true;

else

return false;

}

For example, you might test to see whether an e-mail message contains the text “ c++” with the call containsKeyword(emailObj, " c++");.

1. (10 pts) **OOA & OOD: Draw the UML Class Diagram.**

|  |
| --- |
| Email |
| - sender: string //1  - recipient: string //2  - title: string //3 |
| + Email() //1  + Email(string , string , string , string) //2  + ~Email() //3  + getSender(): string //4  + getRecipient(): string //5  + getTitle(): string //6  + setSender(string): void //7  + setRecipient(string): void //8  + setTitle(string): void //9  + operator =(Email&): Email& //10 |

**ANSWER:**

|  |
| --- |
| Document |
| - getText(): string //1  - setText(string): void //2 |
| + Document(); //1  + ~Document(); //2  + Document(string); //3  + getText(): string //4  + setText(string): void //5  + operator =(Document&): Document& //6 |

|  |
| --- |
| File |
| - getText(): string //1  - setText(string): void //2 |
| + File () //1  + File (string , string) //2  + ~ File () //3  + getPathName(): string //4  + setPathName(string): void //5  + operator =( File &): File & //6 |

1. (50 pts) **Implementation: Visual Studio 2010 AS HOMEWORK11 build and run, Source Code and Screenshot. Submit zipped project to BB!**

[**http://media.pearsoncmg.com/aw/aw\_savitch\_abc\_4/videos/video14\_1/video.html**](http://media.pearsoncmg.com/aw/aw_savitch_abc_4/videos/video14_1/video.html)

**ANSWER:**



