# SAVITCH ABSOLUTE C++

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

**Homework 10 - On Lecture 12 – Classes**

**(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 Homework10.doc & Homework10.zip**

**in BB?**

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

1. **DID TURN IN HOMEWORK INSTRUCTIONS? \* -10 points**
2. **1.? H & E (attach to BB) – WORD ONLY 30 points**
3. **2.? 30 points**
4. **3.? 40 points**
5. **Homework10.1 C++.zip NOT submitted to BB? \* -20 points**
6. **Homework10.1 C++.zip NOT running in class? \* -20 points**
7. **Homework10.2 C++.zip NOT submitted to BB? \* -20 points**
8. **Homework10.2 C++.zip NOT running in class? \* -20 points**
9. **Homework10.3 C++.zip NOT submitted to BB? \* -30 points**
10. **Homework10.3 C++.zip NOT running in class? \* -30 points**

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

TA **grade or check**

**By Hand?**

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

# 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!

**Screenshot?**

**1.** (30 pts) **UML Class Diagram** (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**).

Fraction

Define a Class for a type called Fraction. This Class is used to represent a ratio of two integers. Include mutator functions that allow the user to set the **private** numerator and the **private** denominator. Also include a member function that returns the value of the numerator divided by the denominator as a double. Include an additional member function that outputs the value of the fraction reduced to lowest terms. For example, instead of outputting 20/ 60 the function should output 1/ 3. This will require finding the greatest common divisor for the numerator and denominator, and then dividing both by that number. Also create a test program that contains main member function.

Fraction:3: getSimpleFraction(): String

Fraction:3: getValue(): double

Fraction:2: denomerator: int

Fraction:1: numerator: int

Fraction:2: setDenomerator(int): void

Fraction:1: setNumerator(int): void

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

|  |
| --- |
| Main |
|  |
| + main(int , char\*): int //1 |

**ANSWER:**

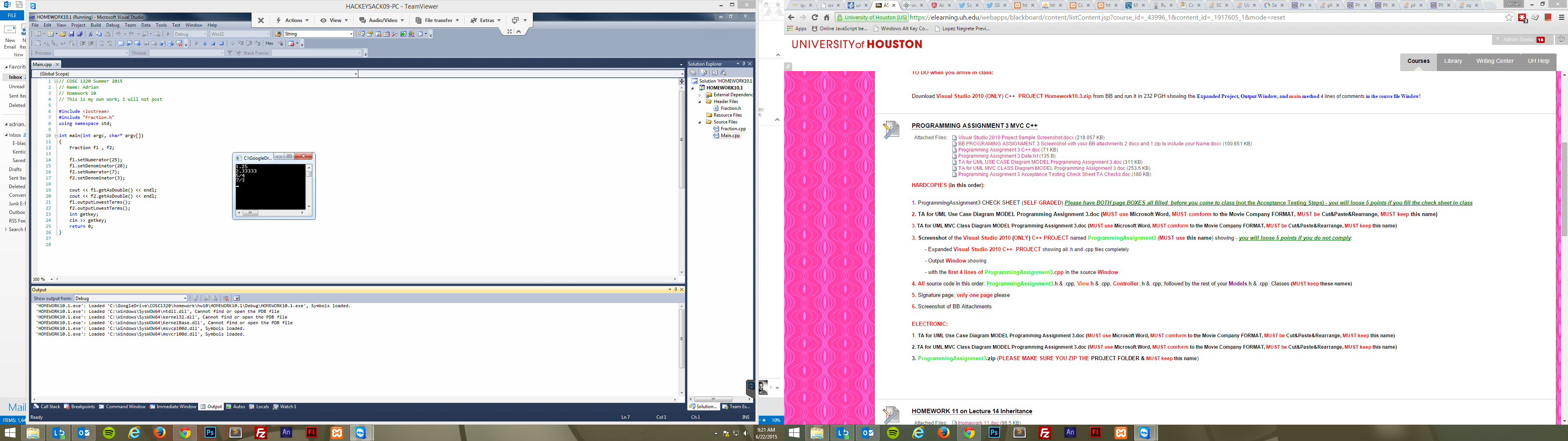
Uses

|  |
| --- |
| Fraction |
| - numerator: int //1  - denomerator: int //2 |
| + setNumerator(int): void //1  + setDenomerator(int): void //2  + getSimpleFraction(): String //3 |

1. (20 pts) **Implementation: Visual Studio 2010 AS HOMEWORK10.1 build and run, Source Code and Screenshot. Submit zipped project to BB!**

<http://media.pearsoncmg.com/aw/aw_savitch_abc_4/videos/video6_1/video.html>

**ANSWER:**



**2.** (30 pts) **UML Class Diagram** (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**).

Enter and run with debugger option BankAccount Class Display 7.4 from page 291 by using a separate compilation with the following files: BankAccount.h BankAccount.cpp and Driver.cpp.

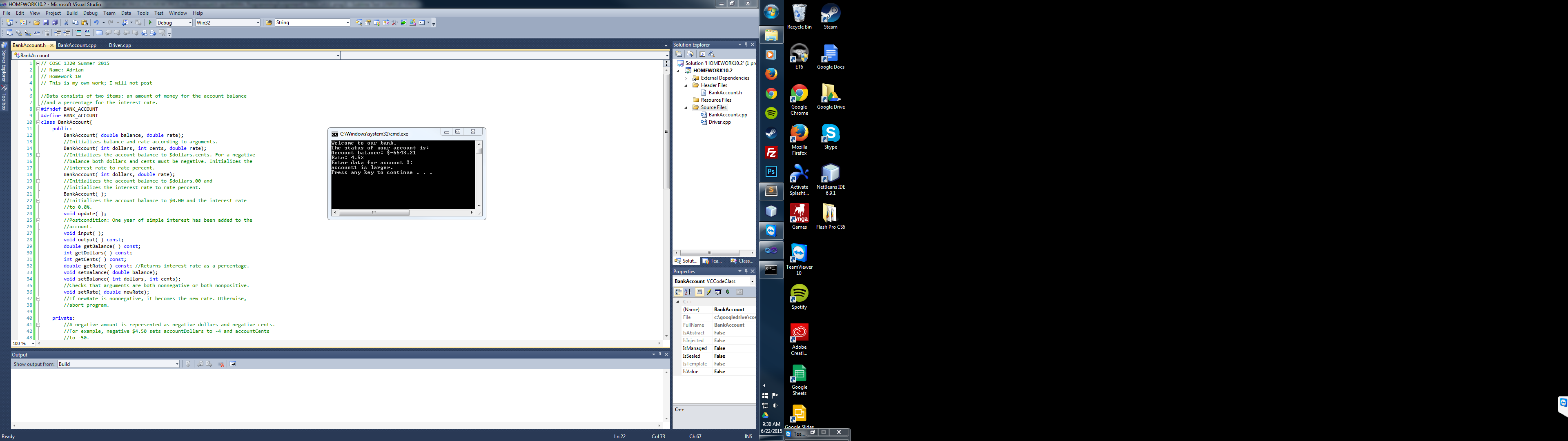
1. (10 pts) **OOA & OOD: Draw the UML Class Diagram.**

**ANSWER:**

|  |
| --- |
| BankAccount |
| - accountDollars: int //1  - accountCents: int //2  - rate: double //3 |
| + BankAccount( double balance, double rate) //1  + BankAccount( int dollars, int cents, double rate) //2  + BankAccount( int dollars, double rate) //3  + BankAccount( ) //4  + update( ): void //5  + input( ): void //6  + output( ) const: void //7  + getBalance( ) const: double //8  + getDollars( ) const: int //9  + getCents( ) const: int //10  + getRate( ) const: double //11  + setBalance( double balance): void //12  + setBalance( int dollars, int cents): void //13  + setRate( double newRate): void //14  - dollarsPart( double amount) const: int //15  - centsPart( double amount) const: int //16  - round( double number) const: int //17  - fraction( double percent) const: double //18 |

1. (20 pts) **Implementation: Visual Studio 2010 AS HOMEWORK10.2 build and run, Source Code and Screenshot. Submit zipped project to BB!**

**ANSWER:**



**3.** (40 pts) **UML Class Diagram** (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**).

You operate several hot dog stands distributed throughout town. Define a Class named HotDogStand that has a **private** member variable for the hot dog stand’s ID number and a **private** member variable for how many hot dogs the stand sold that day. Create a Constructor that allows a user of the Class to initialize both values. Also create a function named JustSold that increments the number of hot dogs the stand has sold by one. This function will be invoked each time the stand sells a hot dog so that you can track the total number of hot dogs sold by the stand. Add another function that returns the number of hot dogs sold. Finally, add a **static** variable that tracks the total number of hot dogs sold by all hot dog stands and a **static** function that returns the value in this variable. Write a separate main function to test your Class with at least three hot dog stands that each sell a variety of hot dogs.

HotDogStand: 5: getNumSoldAllStands():int

HotDogStand: 1: HotDogStand()

HotDogStand: 2: HotDogStand( int , int)

HotDogStand: 1: id:int

HotDogStand: 2: soldToday:int

HotDogStand: 3: JustSold():void

HotDogStand: 3: soldAllStands: int

HotDogStand: 4: getNumSold():int

HotDogStand

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

**ANSWER:**

|  |
| --- |
| Driver |
|  |
| + main(): int //1 |

|  |
| --- |
| HotDogStand |
| - id:int //1  - soldToday:int //2  - soldAllStands: int //3 |
| + HotDogStand() //1  + HotDogStand( int , int) //2  + JustSold():void //3  + getNumSold():int //4  + getNumSoldAllStands():int //5 |

1. (30 pts) **Implementation: Visual Studio 2010 AS HOMEWORK10.3 build and run, Source Code and Screenshot. Submit zipped project to BB!**

<http://media.pearsoncmg.com/aw/aw_savitch_abc_4/videos/video7_2/video.html>

**ANSWER:**

