

**FACULTY OF COMPUTERS AND INFORMATION,
CAIRO UNIVERSITY**

CS213: Programming II
Year 2018-2019
First Semester

Assignment 2 – Version 2.0

Course Instructors:
Dr. Mohammad El-Ramly
Dr. Mohammad Nassef

Revision History

Version 1.0	By Dr Mohammed El-Ramly	10 October 2018	Main Doc
Version 2.0	By Dr Mohammed El-Ramly	15 October 2018	Final Doc

CS111: Fundamentals of CS

Assignment 2 (7 marks) – Version 2.0



Cairo University, Faculty of Computers
and Information

Objectives

This assignment aims to:

- 1- Train students on creating and using classes, objects and OOP in C++.
- 2- Train students on program modularization and separate compilation.
- 3- Train students on team work and cooperation.
- 4- Train students on the role of programming in society and developing a software business.

Instructions

1. This is a big assignment, **START WORKING YESTERDAY**. لا تنتظر لآخر لحظة و ابدأ الان.
2. These instructions must be followed to get the full mark. يجب اتباع هذه التعليمات بكل دقة.
3. **Deadline is Monday 29th of October 2018 @ 11:59 pm. Weight is 7 marks.**
4. **Work with a team from the same group.** Team consists of three students whose IDs do not end with the same digit. For example, 2017023, 20170433 and 20170124 cannot be in one team because two of them have IDs ending with 3. الفريق من ٣ طلاب لا ينتهي رقم بطاقة الكلية لهم بنفس الرقم 3.
5. If you have any problem or difficulty, ask your TA or the professor. **We love to help you.**
6. Please submit **only work that you did yourself**. If you copy work from your friend or book or the net **you will fail the course**. تسليم حلول منقولة من أى مصدر يؤدي إلى الرسوب في هذا المقرر. لا تغش الحل أو تنقله من أى مصدر و اسألنى فى أى شئ لا تفهمه لكن لا تنقل الحلول من النت أو من زملائك أو أى مكان

Task 1 (2 marks) – C++ Classes and Separate Compilation

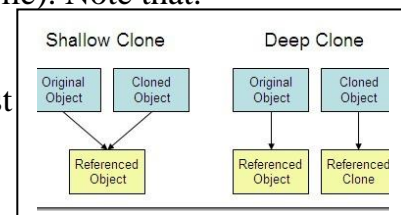
This task will continue on Task 1 of Assignment 1. The code you created there will be reused here. **If you did not solve Assignment 1 or only did part of it, you still can work on this task by writing new code to implement the solution for this task here.**

Your team will convert the matrix structure of Assignment 1 **into a class**. Elements inside the structure will be private data members of the class. All the functions you created will be public methods in the class. You will constructors (instead of create matrix, and including a copy constructor that copies a matrix to another), setters and getters and the following methods:

```
Matrix& operator= (matrix mat); // Copy mat content to current mat  
// And return a copy of it
```

Code will be divided into three files **xxx.hpp** (matrix header file) and **xxx.cpp** (matrix implementation file) and **yyy.cpp** (which is demo or main application file). Note that:

1. Assignment (**operator=**) and copy constructor should do **deep copying** by creating another storage and copying content from the first one to the second one. Shallow copying will only copy the address of the data and will make two objects point to the same data storage.
2. All the methods will need one less parameter since the first parameter will be the current object.
3. Output operator (**operator<<**) will become a **friend or free** methods since its first parameter must be of type **ostream**. The same is for input operator (**operator>>**).



CS111: Fundamentals of CS

Assignment 2 (7 marks) – Version 2.0



Cairo University, Faculty of Computers
and Information

Team leader will create the matrix class and each one should convert and test his own functions he created for Assignment 1. Then team should integrate the different functions in one program and write a demo program to demonstrate the use of these functions.

Task 2 (1.5 marks)

1. Each team member will solve one of the following problems according to their ID.
2. Some problems are taken from this book: **Problem Solving with C++**, ninth edition and **Programming Abstractions in C++**, 2012.
3. Each student **must solve the correct problem** according to the following: IDs ending with X will solve the corresponding problem, where X is:



0. Problem 3 on page 311
(Programming Abstractions)
1. Problem 9 on page 315
(Programming Abstractions)
2. Problem 13 on page 891 (Problem Solving)
3. Problem 8 on page 615 (Problem Solving)
4. Problem 11 on page 617 (Problem Solving)
5. Problem 2 on page 884 (Problem Solving)
6. Problem 4 on page 312 (Programming Abstractions)
7. Consider the stack classes created in Lecture 3 and included in file Lecture3Code-CandOOPStacks. Rewrite this class using a **valarray** for storing the data. Add methods **void clear ()**, **bool isEmpty ()** and **int getSize ()**.
The first erases the stack content and makes size = 0. The second returns true if the stack is empty. The third returns the number of the items currently in the stack. Write an application to test the functions of the stack. Use separate compilation and separate interface from implementation from application.
8. Define a class hierarchy whose root is class **Shape**. **Shape** is an abstract class. A shape has a color, a constructor and a methods **getColor** and **setColor**. There are two kinds of shapes that inherit from class **Shape**: 2D shapes and 3D shapes. 2D and 3D shapes are also abstract classes. 2D shapes have a virtual method: **area**. 3D shapes have virtual methods: **surfaceArea** and **volume**. Create classes **Circle**, **Square**, **Sphere** and **Cylinder** that inherit from these two classes. Inherited classes should have the necessary attributes and should override the virtual methods. Write a small application to test these classes. **Separate the interface from the implementation from the application.**
9. Problem 3 on page 696 (Problem Solving)

CS111: Fundamentals of CS

Assignment 2 (7 marks) – Version 2.0



Cairo University, Faculty of Computers
and Information

Task 3 (1.5 marks)

1. **All the team must work together on this task.**
2. Different variations of `int` and `float` exist in C++ and other languages. They are limited by minimum and maximum values. Sometimes it is desired to have versions of these types with unlimited bounds. Java solves this problem by providing `BigInteger` and `BigDecimal` classes. In this problem it is required to develop a new type that can hold unlimited decimal integer values and performs arithmetic operations on them. You will develop in C++ a class, `BigDecimalInt`, that supports writing statements with extremely long integer values like these:

```
BigDecimalInt num1("123456789012345678901234567890");
BigDecimalInt num2("113456789011345678901134567890");
BigDecimalInt num3 = num2 + num1;
BigDecimalInt num4 = num2 - num1;
cout << "num1 = " << num1 << endl;
cout << "num2 = " << num2 << endl;
//236913578023691357802369135780
cout << "num2 + num1 = " << num3 << endl;
//10000000001000000000100000000
cout << "num2 - num1 = " << num4 << endl;

BigDecimalInt num5("33333333333333333333333333333333");
BigDecimalInt num6("-11111111111111111111111111111111");
BigDecimalInt num7 = num5 + num6;
BigDecimalInt num8 = num5 - num6;
//22222222222222222222222222222222
cout << "num5 + num6 = " << num7 << endl;
//44444444444444444444444444444444
cout << "num5 - num6 = " << num8 << endl;
```

Your task is:

3. Design the class `BigDecimalInt` that has the following public interface (set of operations available to use by developers using the class):

```
BigDecimalInt (string decStr); // Initialize from string
BigDecimalInt (int decInt); // Initialize from integer
BigDecimalInt operator+ (BigDecimalInt anotherDec);
BigDecimalInt operator- (BigDecimalInt anotherDec);
```

4. You will also need to overwrite the `<<` operator.
5. Using data encapsulation, you are free to store the digits of the big decimal integer. You might store them in an array, a vector, a string or whatever. These are details that are not important to the user of your class. You need also to store the sign of the long integer. You will need to build `+` and `-` operations to perform on the representation you chose.
6. Implement the class `BigDecimalInt` and write five test cases (including `-ve` numbers) to test it. Implement a program that runs the test cases and verifies the result.
7. Use separate compilation to separate interface from implementation from application.

CS111: Fundamentals of CS

Assignment 2 (7 marks) – Version 2.0



Cairo University, Faculty of Computers
and Information

Task 4 (1.5 mark) – Starting a Software Business

Software is created to solve real world problems and help people better achieve their tasks. This can be a mobile app, a Web site, a desktop application, an embedded system, etc. For example,

- 1- Skype, Whatsapp, Facebook Messenger, line, etc., help people communicate easily.
- 2- Google Chrome, Internet Explorer, Firefox, etc., help people access web pages on a web server.
- 3- Gym management systems helps gym owners manage membership and gym services
- 4- Steps tracking mobile apps help you know how many steps you walked and stay healthy.
- 5- Auto cruising software help car driver drive the car automatically with minimal manual actions.
- 6- Google translate and similar software help people understand text in a different language.

In this task, develop an idea for a software system based on a real need for a real client. You will collect and document the requirement specifications of this software in details. To do this task:

- 1- Identify a real possible client (or a target group of people) in your personal circle; someone whose work or life will be better and who would be faster software helps him or her.
 - 2- **The group will meet** the client at least twice and will **take a photo of the meeting**.
 - 3- The group will ask the client in details about what he does (they do) and how a software system can help him or her be more efficient.
 - 4- The group will write in ع or E (1) Overview of the required software (2) a list of requirements (all things that software should do) and (3) will draw the main screen(s) of the app using Word or pen or using programs like: <https://balsamiq.com/> (Or search for Mockup apps for Android)
 - 5- The group will submit one report of three pages at least, font 12, single space with their names and IDs, the project idea or title and the specifications of the project as described in 4.
- ١- ابحث في دائرة معارفك عن عميل محتمل يمكن أن يتحسن عملهم بواسطة برنامج على الحاسب (أو مجموعة من الناس مثلاً الشباب أو المسنين أو هواة الرياضة أو الباحثين عن عمل إلخ و يمكن أن يفيدهم عمل برنامج خاص لهم)
- ٢- ستلتقي مجموعتك بالعميل المحتمل (أو المجموعة) مرتين على الأقل و يجب أخذ صورة لهذه اللقاءات.
- ٣- مجموعتك ستسأل العميل عما يفعل و كيف يمكن للبرنامج المحتمل تسهيل مهمته و عمله أو تيسير حياته أو ملء حاجة لديه
- ٤- ستكتب المجموعة بأى لغة (١) مقدمة عامة عن البرنامج المطلوب ووظيفته و (٢) قائمة تفصيلية مطولة بالوظائف المطلوبة في هذا البرنامج و ما ينبغي أن يعمل و (٣) عدة رسومات للشاشات الرئيسية للبرنامج مرسومة باليد أو أى برنامج مثل <https://balsamiq.com/> أو غيره.
- ٥- ستقدم المجموعة تقريراً من ٣ صفحات على الأقل بصيغة بى دى أف PDF مكتوبة بخط مقاس ١٢ و مسافة واحدة بين السطور و عليه اسماء الطلاب و اسم المشروع و التفاصيل المذكورة في ٤.

Task 5 (0.5 mark) – Writing Good Quality Code

No program stays the same. It will need to change to fix bugs, add new features, etc. So, it is very important to write high quality readable code so that you or other developers are able to review and modify this code in the future. In this task, you will:

- 1- Add a header to your program saying who the author is, the purpose of the program, etc.
- 2- Add a header for every function explaining what it does, what parameters it takes and what value it returns.
- 3- Write the code following C++ coding style. <http://geosoft.no/development/cppstyle.html>
- 4- Add comments to any part that is difficult to understand.

CS111: Fundamentals of CS

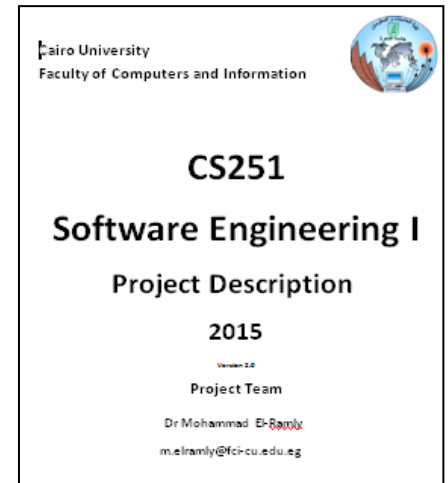
Assignment 2 (7 marks) – Version 2.0



Cairo University, Faculty of Computers
and Information

Submission Instructions

1. Team will submit into **acadox** the following:
 - A zip file with a pdf document with their names and IDs with a cover page like this **and report/photos of Task 4**.
 - The source code of each program in a separate folder.
 - A screen shot for every **GitHub** account.
2. **Zip file name must be:**
FCI-CS213-A2-2018-TAName-GroupNum-ID1-ID2-ID3.zip
3. Team will create a project in **GitHub** to upload code there.
4. Each team member will work individually on his part. **But the team must provide ONE integrated and working program for tasks 1 and 3**.
5. Team members are expected to help each other but not do work of others.
6. **All team members must understand the details** of all programs and be able to explain it or even modify it if needed.
7. Team members are responsible of testing all the programs and making sure they work.
8. TA can ask any team member about any of the programs developed and its code.



Marking Criterion

1. 1.0 for individually developing the part of matrix class of each team member.
2. 0.5 for integrating the different parts in one working program and testing it.
3. 0.5 for dividing the code into header, application and implementation files.
4. 1.5 for writing **an original working program that covers most** of the functionality of Task 2
5. 1.0 for developing **an original working** class **BigDecimalInt** and testing it.
6. 0.5 for evidences of working together and cooperation.
7. Zero for any team member that did not cooperate on doing this task.
8. 1.0 for good quality description of the software project and its requirements
9. 0.5 for drawing of the expected screens of the software
10. -0.5 for any team member not appearing in the photos with the client.
(exemption is given for ladies who prefer not to be photographed)
11. 0.5 for writing good quality code using coding style and adding headers and comments
12. -0.5 for not putting the code in a project in GitHub.
13. -0.5 for not naming the file properly.