

# ASSIGNMENT 02

## DATA STRUCTURE FOR COMPLEX NUMBERS

DS286.Aug16 Data Structures and Programming

August 29, 2016

Submission is due on or before **Sunday, September 11, 2016, 11:59pm IST.**

The assignment carries **50 points**, which is 5% of the course weightage.

### 1 Question

You are required to implement a data structure for handling arithmetic operations on complex numbers. Specifically, the list of operations that should be supported by this data structure is as follows:

1. Adding 2 complex numbers
2. Subtracting 2 complex numbers
3. Multiplying 2 complex numbers
4. Division of one complex number by another

$$x + iy = \frac{a + ib}{c + id}$$

5. Exponentiation of one complex number by another

$$x + iy = (a + ib)^{(c+id)}$$

You are given a basic skeleton program. The initial complex number value present in the program is  $0 + i0$ , and math operations are performed sequentially to update this value by using the `Complex` number object. Your task is to complete the skeleton program in the places indicated. You cannot use any header files besides the ones already included in the skeleton.

You are also required to update the code in the `main()` method to implement test cases. These tests should comprehensively verify if your `Complex` class works as expected for all valid and invalid samples.

## 2 Test Cases

Check the `input` file provided for sample inputs. Update the skeleton code, compile and run the program for the given inputs using the following commands:

```
make clean
make
./complexdemo.out < input
...
```

### Sample inputs:

```
./complexdemo.out
: + 2 4
: - 1 2
: q 0 0
= 1 + i2
```

```
./complexdemo.out
: + 2 3
: * 4 5
: q 0 0
= -7 + i22
```

```
./complexdemo.out
: + 4 2
: / 3 -1
: q 1 1
= 1 + i
```

```
./complexdemo.out
: + 7 6
: ^ 2 -8
: q 1 1
= -19674.9 + i14814.3
```

```
./complexdemo.out
: - 2.1 -6.4
: + 5 5
: * 3.6 7.1
: q 1 1
= -70.5 + i61.63
```

```
./complexdemo.out
: t 0 0
Running tests...
```

```
1. Unit tests: [PASSED]
2. Integration tests: [PASSED]
All tests were successful :-)
: q 0 0
= 0 + i0
```

### 3 Submission Instruction

Please follow these instructions carefully. We use automated scripts for evaluation. So a failure to follow these instructions will mean that your submission will not be evaluated.

- Write your program by completing the skeleton code that is provided. Do not modify the original skeleton itself. *Just add extra code to complete the assignment in the space indicated.*
- Put all your files including source file, executable file and *Makefiles*, in a single folder whose name is determined as follows. My full name is “Prateeksha Varshney” where “Prateeksha” is my first name, so the folder name should be `02Prateeksha` for my submission. Please note the capitalization of first letter of the first name.
- This folder should be compressed using the `tar` program as follows:  
`tar -cvf 02Prateeksha.tar 02Prateeksha/`

Note: Any other compression format *will not be accepted* and will be treated as no submission. Its your responsibility to check if the file can be properly uncompressed and all files inside are intact.

- Send a separate mail to the TA’s email address `prateeksha@grads.cds.iisc.ac.in` with the subject line `DS286.Aug16.A02`. Do not write anything more or less in the subject line. Do add any text in the body. Do not send the assignment as a reply-email to any other mail.
- **Only one submission will be accepted.** If multiple emails or files are received, **only the first one** will be taken as the submission. Only the submission received **before the deadline** will be accepted.
- Use only the C++ language for completing this assignment. Make sure the code compiles and executes correctly on the head node of the `turing.cds.iisc.ac.in` server using `g++` command. The code will be compiled and tested on this machine.

### 4 Ethics

You should not get assistance from other students or external sources in directly solving the assignment. Getting help on generic C++ and data structures con-

cepts, e.g., on using lists, strings, libraries, compilation, etc. is accepted. You are encouraged to post questions to the course mailing list so that the TA, instructors or other students can respond. This also ensures you do not have an unfair advantage/disadvantage over other students. If you have received assistance from other sources, send a *separate email to the Instructor and the TA* disclosing the external sources and type of support received.

By making a submission, you are asserting that all code that you submit was designed and developed by you. Do NOT copy and paste code from anyone else! All code will be verified using a plagiarism checker, and *penalties will be imposed* if plagiarism is found from unattributed sources.