

National University Of Computer and Emerging Science
Lahore Campus
Object Oriented Programming

Assignment # 3

Due Date: Wed, May 26th by 11:55pm..

Dear students we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.

1. Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.).
2. Run and test your program on a lab machine before submission.
3. Rename the folder as ROLL-NUM_SECTION (e.g. 20i-0001_A) and compress the folder as a zip file. (e.g. 20i-0001_B.zip). do not submit .rar file.
4. Submit the .zip file on Google Classroom within the deadline.
5. Submission other than Google classroom (e.g. email etc.) will not be accepted.
6. The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.
7. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code. Five Bonus marks will be awarded to well commented/indented code (for all questions).
8. Total Marks: 100
9. If there is a syntax error in code, zero marks will be awarded in that part of assignment.
10. Your code must be generic

Overflow Management

The native int type in C++ cannot store large integers of more than 10 decimal digits. In this project, you will implement a bigint class that behaves like the int type, but can handle unsigned integers of up to 45 digits.

- [illegible]

Submission

Zip all of your files for this assignment, which should include at least four files `bigint.h`, `bigint.cpp`, `driver.cpp` and `bugFixingLog.txt`. Name the zip file `RegNo-ame.zip` and submit the zip file in google classroom.

Grading Policy

1. Program with no compiler error (20 points)
2. Log for fixed compiler bugs (10 points)
3. The default constructor and the overloaded « operator (20 points)
4. Other four constructors (3 points each, 12 points total)
5. Six comparison operators (2 points each, 12 points total)
6. + and - operators (8 points each, 16 point total)
7. The » operator (10 points)

Hints

1. You should then implement the default construct, and write a simple operator«() function that simply prints out the values of v[0], v[1], v[2], v[3], v[4].
 2. After that you can implement the functions in the following order: the « operator, other constructors, the comparison operators, the arithmetic operators, and the » operator. Remember to compile and run the program at least once everytime you add one function.
 3. To implement the « operator, you will need to output potentially 5 integers. You must use setw(9) and setfill('0') to output each value besides the most significant non-zero integer. You must have 'include <iomanip>' in the file in order to use these functions. The following line is an example to use these functions.
s « setfile('0') « setw(9) « vv.v[j];
 4. To implement the » operator, you can first read the string of digits into a character array and then manually convert it into a bigint number.
 5. To implement the + and − operators, you must pay special attention to the carry bit in the operations.
-

Honor Policy

This task is a learning opportunity that will be evaluated based on your ability. **Plagiarized reports or code will get a zero.** If in doubt, ask the course instructor.