

MCA 1st SEMESTER
Object Oriented Prog. Methodology
PRACTICAL SHEET : S2
Submission : 10th January 2022, 3.00 pm

1. Create a class **Rectangle**. The class has attributes **length** and **width**, each of which defaults to 1. It has methods that calculate the **perimeter** and the **area** of the rectangle. It has *set* and *get* methods for both **length** and **width**. The *set* method should verify that **length** and **width** are each floating-point numbers larger than 0.0 and less than 20.0.
2. Create a more sophisticated **Rectangle** class than the one you created in Problem 1. This class stores only the Cartesian coordinates of the four corners of the rectangle. The constructor calls a *set* method that accepts four sets of coordinates and verifies that each of these is in the first quadrant with no single x or y coordinate larger than 20.0. The *set* method also verifies that the supplied coordinates do, in fact, specify a rectangle. Member methods calculate the **length**, **width**, **perimeter** and **area**. The length is the larger of the two dimensions. Include a predicate method **square** that determines if the rectangle is a square.
3. Create a class **HugeInteger** that uses a 40-element array of digits to store integers as large as 40-digits each. Provide member methods **inputHugeInteger**, **outputHugeInteger**, **addHugeIntegers** and **subtractHugeIntegers**. For comparing **HugeInteger** objects, provide methods **isEqualTo**, **isNotEqualTo**, **isGreaterThan**, **isLessThan**, **isGreaterThanOrEqualTo** and **isLessThanOrEqualTo**--each of these is a “predicate” method that simply returns **true** if the relationship holds between the two huge integers and returns **false** if the relationship does not hold. Also provide a predicate method **isZero**.
4. Write a program that demonstrates the use of all types of Constructors and a destructor.
5. Write a Program and demonstrate whether the use of inline function increases or decreases the size of the object code.
6. Overload the subscript operator [] to return the largest element of a collection, the second largest, the third largest, etc.
7. Create a class **RationalNumber** (fractions) with the following capabilities:
 - (a). Create a constructor that prevents a 0 denominator in a fraction, reduces or simplifies fractions that are not in reduced form and avoids negative denominators.
 - (b) Overload the addition, subtraction, multiplication and division operators for this class.
 - (c) Overload the relational and equality operators for this class.
8. For an object m1 of matrix class, can we use m1<<cout . If yes prove it.
9. Overload the *,+,-,=,!= and = operators for the complex class.
10. For the class Date with properties int month; int day; int year, overload the following operators.
 - a. + operator [a+b (a is of date type and b is an integer), use the assumption that all years have 360 days and months 30 days.
 - b. – operator [a-b(same as above)]
 - c. = operator [supporting a=b=c=d]
 - d. <,<=,>,>=
 - e. ++, --[post and pre both]