

## Lab 9

### Ques 1

Create a class **ProblemSolution** with following characteristics

- First constructor accepts two integer parameters N1 and N2 and displays addition of 2 numbers.
- Second constructor accepts three integer parameters N1, N2 and N3 and displays addition of 3 numbers.

#### Input

10  
20  
45

where

- First line represents a value of N1.
- Second line represents a value of N2.
- Thrid line represents a value N3.

#### Output

30  
75

### Ques 2

Define a class named **Rectangle** with following characteristics:

- Data members
  - double field **x** and **y** that specifies the center of the rectangle.
  - integer field **width** and **height** of the rectangle.
- Methods
  - The default constructor creates the default rectangle centered at (0,0) and width and height as 1.

- Another parameterized constructor has 4 parameters representing x, y, height and width values.
- `getArea()` - Returns area of rectangle.
- `getPerimeter()` - Returns perimeter of rectangle.
- `contains(double x, double y)` - returns true if the specified point (x,y) is inside the current rectangle.

#### Input

2.0  
2.0  
10  
16  
10.0  
7.0

where,

- First two lines represent value **x** and **y** respectively which is a center of the custom rectangle.
- Third and fourth lines represent **height** and **width** of the custom rectangle respectively.
- Fifth and sixth lines represent **x** and **y** coordinates of a point to be which is to be checked.

#### Output

4  
1  
true  
52  
160  
true

where,

- First three lines represent perimeter, area, and result of whether the point (0.5, 0.1) is inside the default rectangle. (this value will not change with change in test case)
- Fourth and fifth lines represent perimeter and area, respectively, of the custom rectangle created from user input.
- Sixth line represents whether given point is inside the custom rectangle.

Ques 3

Create a **desctructor** of class ProblemSolution which will print "Destructor called".

#### Output

Destructor called

#### Ques 4

Create a class named **Time** with following characteristics:

1. It has private parameters **hour, minute,second**.
2. Create a constructor which takes total time of day in seconds, converts into hour, minute and second and store in the respective private vairables. Use **long** data type as input argument.
3. Create a method **getTime()** which returns time in 24- hour format (HH:MM:SS) as string.