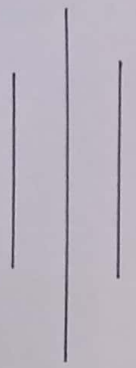
KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE



A

Lab Report On

Object Oriented Programming {COMP116}

Lab Report No:- 2

Submitted by:

Ashraya Kadel

CE : Batch 2022

RollNo: 25

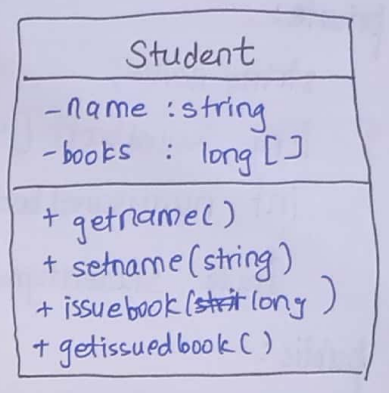
Submitted to.

Rajani Chulyadyo

Department of Computer Science and Engineering

SUBMISSION DATE: 15/09/2023

<Q.1>: Implement a class called student, shown in the class diagram.



+Data members:
→ Name of student
→ Array of book ID's
+ Member functions
→ getname() returns name
→ setname() sets name
→ issuebook(long) adds book ID's
→ getissued book() returns ID of books borrowed.

Ans

x) Source Code:

```
#include <iostream>
```

<Q.2>: Define an enum called Grade, with following values: A, A-, B, B-, C, C+, C, C-, D and F. Modify Student class of Q.1 to store the grade of student by adding getter and setter.

Ans

(x): Source Code:

```
#include <iostream>
#include <string>
enum Grade {
  A, A_M, B_P, B, B_M, C_P, C, C_M, D, F;
const char * gradeletters[] = { "A", "A-", "B+",
                                "B", "B-", "C+", "C",
                                "C-", "D", "F" };
```

```
class Student {
  private:
    string name;
    long issuedbooks [10];
    int numissuedbooks = 0
    Grade studentgrade;
  public:
    void setname (const string & newname)
    { name = newname; }
    string getname const { return name; }
    void issuebook (long bookID) {
        if (numissuedbooks < 10)
        { issuedbooks [numissuedbooks] = bookID;
          numissuedbooks++; }
        else {
        std:: cout << " Maximum numbes of books reached!.";
        } }
    int getbookscount() const { return numissuedbooks; }
    long getIssuedBooks (int i) {
      if (i>=0 && i< numissuedbooks)
      { return issuedbooks [i]; }
      return -1; }
```



```
    void setgrade (const string & grade)
  { studentgrade = grade; }
    string getgrade const { return studentgrade; }
};

int main ( ) {
    double percentage;
    Student student_1;
    student_1. setname ("Ashraya Kadel");
    student_1. issuedbook (1011);
    student_1. issuedbook (2012);
    std:: cout << "Enter of percentage" <<std:: endl;
    std:: cin >> ~~dou~~ persentage;
    Grade sgrade;
    if (percentage >=90) { sgrade = A; }
  elseif (percentage < 90 && percentage >=85)
  { sgrade = A-m; }
  elseif (persentage < 85 && percentage >=80)
  { sgrade = B-p; }
  elseif (percentage < 80 && percentage >= 75)
  { sgrade = B; }
  elseif (percentage < 75 && percentage >=70)
  { sgrade = B-m; }
  else if (persentage < 70 && persentage >=65)
  { sgrade = C-p; }
```

```
elseif (percentage <65 && percentage >=60 )
  {sgrade = C; }
elseif (percentage < 60 && percentage >= 55 )
  {sgrade = C-m; }
elseif (percentage < 55 && percentage >= 50)
  {sgrade = D; }
else
  {sgrade = F; }

student_1.setname (sgrade);

std:: cout << " Student name :" << student_1.getname()
    << std:: endl;
std:: cout << " Issued books are" <<std::endl;
for (int i=0; i<student_1.getbookscount(); i++)
{  std:: cout<< student_1.getIssuedBooks(i) <<std::endl; }
std:: cout << "  " << std:: endl;
std:: cout << " Grade :  " << student_1.getgrade() <<
 std:: endl;

 return 0;
}
```

x) Output:

```
Percentage 92.086
Student name: Ashraya Kadel
Issued books are
1019
2012
Grade: A
```


x) Description:

This program has a class student, enum Grade and has member functions to set_name, get name, getissued books, counting issued books and returning the grade. This is the combination of qno 1 & 2.

<Q.3>: A rectangle can be defined by two points (top-left and bottom-right or top-right and bottom-left). Implement a class called Rectangle using the Point class we saw during the lecture. The Rectangle class must have the following methods

i) void setPoints (const point&, const point&)
ii) void getDimension (double&, double&)
iii) double perimeter()

Write the main function to check if your implementation works correctly.

Ans:

x) Source Code:

```
#include <iostream>
#include <cmath>

using namespace std;
```

```
class Point {
  private:
    double x;
    double y;
  public:
    Point (double xCoord, double ycoord) : x (xCoord),
                                           y (ycoord) {}
    double getX() const { return x; }
    double getY() const { return y; }
};

class Rectangle {
  private:
    Point topLeft;
    Point bottomRight;

  public:
   Rectangle (const Point& tl, const Point& br) :
              ~~topLeft = tl; bottomRight = br;~~
              topLeft (tl), bottomRight (br) {}

   void setPoints (const Point& tl, const Point& br){
       topLeft = tl; bottomRight = br; }

   void getdimensions (double& width, double& height)
  { width = abs (bottomRight.getX() - topLeft.getX());
    height = abs (topLeft.getY() - bottomRight.getY());
  }
```



```
   double perimeter ()
  { double width, height;
    getdimensions (width, height);
    return 2* (width+ height); }
  };

  int main()
  {
      Point tl (1.0, 4.0);
      Point br (7.0, 1.0);

     Rectangle rect (tl, br);
     rect.setPoints (tl, br);

     double width, height;
     rect.getDimensions (width, height);

     cout << "Rectangle Dimensions: " << endl;
     cout << " Width: " << width << endl;
     cout << " Height: " << height << endl;
     cout << " Perimeter: " << rect.perimeter () << endl;
     return 0;
  }
```

(X) Output:

```
Rectangle Dimensions:
Width     = 6
Height    = 3
Perimeter = 18
```

(X) Description:

This program contains two classes Point and Rectangle. They help us to get two opposing points of a diagonal, ie, top left and bottom right and helps us get the width, height and perimeter