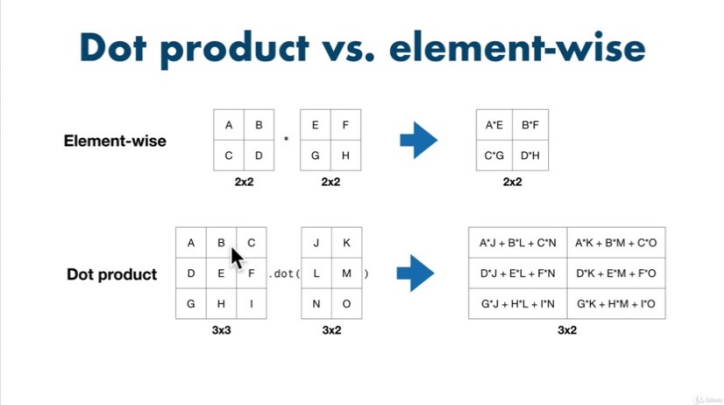
CSCI251 – Assignment 2

Complete ALL questions. Submit the .h and .cpp files to Moodle dropbox by the stated deadline on Moodle.

1. Construct the following class `Matrix` with corresponding .h and .cpp files and develop all methods including constructors and destructors. Download the main.cpp for test cases.

```
class Matrix{
  private:
    int rows, cols;
    double** p;

  public:
    Matrix();     // a default constructor
    Matrix(int, int);    //a parameterized constructor
    Matrix(const Matrix &); // a copy constructor

    void set(int i, int j, double val); //set value at i and j
    double get(int& i, int & j) const; //get value at i and j

    //Assigns (copies) a Matrix.  Return the same
    Matrix& assign(const Matrix &);

    //Adds two Matrices and returns the result
    Matrix& add(const Matrix&);

    //Subtracts two Matrices and returns the result
    Matrix& subtract(const Matrix&);

    //Returns the dot product of two Matrices.  Refer below fig.
    Matrix& multiply(const Matrix&);

    //Element-wise multiplies two Matrices and return the result.
    //Refer below fig.
    Matrix& multiplyElement(const Matrix&);

    //Add a value to all elements in the Matrix
    Matrix& add(double);

    //Multiply all elements in the Matrix by a value
    Matrix& multiply(double);

    //Displays all elements in the Matrix.
    //The result should look like a matrix
    void display();

    //destructor
    ~Matrix();

  };
```



2. Write a class Circle that

```
class Circle{
    private:
        double h, k, radius; // (h, k) = circle coordinate

    public:

        // A default constructor
        // Set the centre and radius of circle as 0.
        Circle();

        // Constructor with the centre of circle.
        // Set the radius as 0.
        Circle (double h, double k);

        // Constructor with the centre of circle and raidus.
        Circle (double h, double k, double r);

        // Set centre of the circle
        void setCentre(double h, double k);

        // Set radius of the circle
        void setRadius(double r);

        // Returns area of the circle
        double getArea();

        // Returns the index of the largest circle within an array of
        // circles. If two circles are in the same size, return the
        // first encountered
        int findLargest(Circle* c_arr, int size);

        // Check whether parameterized circle is within the circle
        // Hint:
        // The smaller circle lies completely inside the bigger circle
        // without touching each other at a point of circumference.
        // If this case happens, the sum of the distance between the
        // centres and radius is lesser than the bigger radius, then
        // obviously the smaller circle lies completely inside the
        // circle, without touching the circumference.

        bool checkWithin(const Circle& c);

        // Return a vector of circles overlapping the largest circle
        Circle** overlapping(Circle* vc, int arrSize);

        //destructor
        ~Circle();
};
```

Submission

Submit your work to Moodle dropbox before deadline stated on Moodle. Do not compress your files.