

Julius Ranoa  
CSC 121 001 Computer Science I  
Programming Finals

### Screenshot of console window

---

Done.

Process finished with exit code 0

---

### Circle.h

---

```
#ifndef FINALS_CIRCLE_H
#define FINALS_CIRCLE_H

class Circle {

private:
    double radius;
    double pi = 3.12159;

public:
    // Constructors
    Circle() { radius = 0; }
    Circle(double r) { radius = r; }

    // Mutators and Accessors
    void setRadius(double r) { radius = r; }
    double getRadius() { return radius; }
    double getArea();
    double getDiameter();
    double getCircumference();

};

#endif //FINALS_CIRCLE_H
```

---

## Circle.cpp

---

```
#include "Circle.h"

double Circle::getArea() {
    return pi * radius * radius;
}

double Circle::getDiameter() {
    return radius * 2;
}

double Circle::getCircumference() {
    return 2 * pi * radius;
}
```

---

## main.cpp – Part 1 of 3

---

```
#include "Circle.h"
#include <iostream>
#include <fstream>
using namespace std;

void TaskOne();
void TaskThree();
int getN(fstream &);
void getData(fstream &, double *, int);

int main() {
    TaskOne();

    // TASK THREE
    // Part I. Reading.
    fstream sourceFile("radius.dat", ios::binary | ios::in);
    if (!sourceFile) {
        cout << "Cannot open file. " << endl;
        return -1;
    }
    int n = getN(sourceFile);
    double *radiusCirclesIn = new double[n];
    getData(sourceFile, radiusCirclesIn, n);

    sourceFile.close();

    // Part II. Saving.
    fstream newFile("circleData.txt", ios::out);
    Circle c;
    if (!newFile) {
        cout << "Cannot create file. " << endl;
        return -1;
    }
}
```

---

## main.cpp – Part 2 of 3

---

```
    for (int i = 0; i < n; i++) {
        c.setRadius(radiusCirclesIn[i]);
        newFile << c.getRadius() << " ";
        newFile << c.getArea() << " ";
        newFile << c.getDiameter() << " ";
        newFile << c.getCircumference();
        newFile << endl;
    }
    newFile.close();

    cout << "Done." << endl;
    return 0;
}

void TaskOne() {
    double radiusCirclesOut[] = {
        10, 20, 30, 40, 50, 60, 70, 80, 90, 100
    };
    fstream file("radius.dat", ios::binary | ios::out);
    if (!file) {
        cout << "Cannot create new file. " << endl;
        exit(0);
    }

    int size = sizeof(radiusCirclesOut) / sizeof(radiusCirclesOut[0]);
    for (int i = 0; i < size; i++) {
        file.write(
            reinterpret_cast<char*>(&radiusCirclesOut[i]), sizeof(double)
        );
    }
    file.close();
}

int getN(fstream &sourceFile) {
    if (!sourceFile) return 0;
    int n = 0;
    double tempRadius;
    while (sourceFile.good()) {
        sourceFile.read(reinterpret_cast<char*>(&tempRadius), sizeof(tempRadius));
        n++;
    }
    n--; // The condition only turns false when the last loop (the first invalid) runs
        // So, this code is a workaround for that.

    // Reset.
    sourceFile.clear();
    sourceFile.seekg(0, ios::beg);

    return n;
}
```

---

### main.cpp – Part 3 of 3.

---

```
void getData(fstream &sourceFile, double *radii, int n) {
    double tempRadius;
    if (!sourceFile) return;
    for (int i = 0; i < n && sourceFile.good(); i++) {
        sourceFile.read(reinterpret_cast<char *>(&tempRadius), sizeof(tempRadius));
        radii[i] = tempRadius;
    }
}
```

---

### radius.dat

*\* The contents below are copied and directly pasted into Word.*

---

\$@4@>@D@I@N@◆Q@T@◆V@Y@

---

### circleData.txt

---

```
10 312.159 20 62.4318
20 1248.64 40 124.864
30 2809.43 60 187.295
40 4994.54 80 249.727
50 7803.97 100 312.159
60 11237.7 120 374.591
70 15295.8 140 437.023
80 19978.2 160 499.454
90 25284.9 180 561.886
100 31215.9 200 624.318
```

---