

Programming Assignment 2: Selection

1 Setting up the Programming Environment

1. To create a new directory (folder) called `hw2` and move to that directory, type:

```
cd
mkdir hw2
cd hw2
```

2. Copy the file `incircles_template.cpp` from the directory `/class/cse1222/9643/hw2/` into the current directory by typing:

```
cp /class/cse1222/9643/hw2/incircles_template.cpp incircles.cpp
```

3. Copy the file `incircles_solution.exe` from the directory `/class/cse1222/9643/hw2/` into the current directory by typing:

```
cp /class/cse1222/9643/hw2/incircles_solution.exe .
```

Be sure to include the period, “.”.

2 Work by Yourself

All lab and programming assignments are to be done by yourself. You may discuss labs or assignments with other students in the class but DO NOT LOOK AT ANYONE’S CODE OTHER THAN YOUR OWN. Needless to say, you should not share or copy anyone else’s code.

3 Programming Assignment

Effective commenting and tabbing will affect your grade. The “style” of your program should follow the style of the sample programs in the course notes. Your program should have the file name, your name, creation and last modification dates and a brief description of the program in the comments at the top of the program. The declaration of every variable should have a comment.

Write a program that reads in the center coordinates and radii of three circles, `Circ1`, `Circ2` and `Circ3`, and reports the location of a query point relative to the circles. The location can be one of eight possibilities: contained in `Circ1`, `Circ2` and `Circ3`, contained in `Circ1` and `Circ2`, contained in `Circ1` and `Circ3`, contained in `Circ2` and `Circ3`, contained in `Circ1`, contained in `Circ2`, contained in `Circ3`, not contained in any circles.

Run `incircles_solution.exe` to see an example of the program. Your program should behave exactly like `incircles_solution.exe` with the same input and output.

1. Prompt and read in the x and y coordinates of the center of circle Circ1;
2. Prompt and read in the radius of circle Circ1;
3. Prompt and read in the x and y coordinates of the center of circle Circ2;
4. Prompt and read in the radius of circle Circ2;
5. Prompt and read in the x and y coordinates of the center of circle Circ3;
6. Prompt and read in the radius of circle Circ3;
7. Prompt and read in the x and y coordinates of the query point;
8. Point (x, y) is contained in the circle with center (cx, cy) and radius r if:

$$\sqrt{dx^2 + dy^2} \leq r.$$

where dx is $x - cx$ and dy is $y - cy$. By applying this equation to each circle and the query point, determine which circles contain the query point.

Note: The operator \wedge is NOT the square operator in C++. Multiply a value with itself to compute its square.

9. Output should be one of the following:

```

None of the three circles contain point (x,y).
Circle Circ1 contains point (x,y).
Circle Circ2 contains point (x,y).
Circle Circ3 contains point (x,y).
Circles Circ2 and Circ3 contain point (x,y).
Circles Circ1 and Circ3 contain point (x,y).
Circles Circ1 and Circ2 contain point (x,y).
Circles Circ1, Circ2, and Circ3 contain point (x,y).

```

In the output x and y should be replaced by the floating point numbers which are the actual coordinates of x and y .

10. Circle center coordinates, query point coordinates and circle radii are floating point numbers. Use double precision.
11. Be sure to modify the header comments “Created by”, “Creation Date”, and “Synopsis” at the top of the file;
12. Be sure that there is a comment documenting each variable;
13. Be sure that your if statements and blocks are properly indented;

14. Test your program on various inputs, including:

- Circ1: center (2,0), radius 3; Circ2: center (0,2), radius 3; Circ3: center (-2,0), radius 3; query point: (0,0).
- Circ1: center (2,0), radius 3; Circ2: center (0,2), radius 3; Circ3: center (-2,0), radius 3; query point: (1,1).
- Circ1: center (2,0), radius 3; Circ2: center (0,2), radius 3; Circ3: center (-2,0), radius 3; query point: (0,-2).
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (0.75,2)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (1.25,2.25)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (0, 1.5)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (1,3)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (0,0)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (-1,6)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (2, 4.75)
- Circ1: center (1.5,0.5), radius 2; Circ2: center (-1,3.5), radius 2.5; Circ3: center (2,3), radius 1.75; query point: (1.25, 4.75)

Check your output against the output from `incircles_solution.exe`.

15. Follow the algorithm steps described above and use this algorithm in the given template to receive full credit.

4 Program Submission

Submit your file `incircles.cpp` in the hw2 drop box on Carmen. DO NOT submit the file `a.out`.

If you do not submit your program, you will receive zero credit for the homework.

If your program does not compile and run you will receive zero credit for the homework.