

# Laboratory 4

*For weeks starting October 25 and November 1.*

Repeat Laboratory 3 but implement your module so that you can store up to 100 different circles, each circle identified by an integer name between 1 and 100 (inclusive). Initially there should be no circles stored. No data is to be stored for circle objects with identifiers less than 1 or greater than 100. The first argument of every program will be the name of the circle “object” or “variable” that the program is operating on.

The first program to be called should be **SETTOL**(eps). The tolerance specified (eps) applies to all circles stored. If a user calls one of the programs, **S3P**, **SCR**, or **SCA**, before calling **SETTOL**(eps), no data on the circle should be stored.

Below you will find the signature of each of the 8 programs. Other than the additional parameter, which identifies the circle object to be used, the meaning of the programs is unchanged from Lab 3.

1. **SETTOL**(eps) - see previous assignments. The programs below will use the value of eps as the tolerance in determining equality of real numbers.

For all of the programs below return -5 if c is out of range and SETTOL has been called. Do this even if another code is specified by the tables in Laboratory 3. There is no data associated with such values of c.

If c is within range but SETTOL has not been called, return -6.

If c is out of range and SETTOL has not been called, return -7.

2. **S3P**(c,  $x_{A0}$ ,  $y_{A0}$ ,  $x_{A1}$ ,  $y_{A1}$ ,  $x_{A2}$ ,  $y_{A2}$ ); c is integer. All other arguments are real; return is integer.

3. **SCR**(c,  $x_{A0}$ ,  $y_{A0}$ , r); c is integer. All other arguments are real; return is integer.

4. **SCA**(c,  $x_{A0}$ ,  $y_{A0}$ , a); c is integer. All other arguments are real; return is integer.

5. **GA**(c, a); c is an integer, a is a real variable; return is integer.

6. **GR**(c, r); c is an integer, r is a real variable; return is integer.

7. **GC**(c, x, y); c is integer. All other arguments are real; return is integer.

8. **INCIRC**(c, x, y); c is integer. All other arguments are real; return is integer.