

Computer Science 102

Lab 9

In this lab you will convert some vector library function to overloaded C++ class methods. You are *not required* to convert the vector components of your raytracer to C++ but you are free to do so if you wish. A sample *main.cpp*, *vec.h* and sample output file *lab9.log* are provided for you. These are *not* guaranteed to represent a complete test of the components.

Your mission is to write a module named *vec.cpp* which will contain the implementations of the following class methods. Overloaded methods such as *vec_diff()* and *vec_dot()* must have *multiple* implementations.

```
class vec_t
{
public:
    vec_t();
    vec_t(double, double, double);

    /* subtract this vector from v2 with the result in v3 */

    void vec_diff(const vec_t v2, vec_t &v3);
    void vec_diff(const vec_t *v2, vec_t &v3);
    void vec_diff(const vec_t &v2, vec_t *v3);

    /* Use cin to read x, y, z components from stdin */

    void vec_read(void);

    /* Print the label, x, y, z and newline to the standard */
    /* error IN THE FORMAT SHOWN in the example output */

    void vec_prn(char *label);

    /* Return the dot product of this this vector with the v2 vector */

    double vec_dot(const vec_t &v2);
    double vec_dot(const vec_t *v2);

    /* multiply x, y, z components of this vector by "fact" */

    void vec_scale(double fact);

    /* Return the length of this vector */

    double vec_len(void);
```

```
private:
    double x;
    double y;
    double z;
};
```

In this lab you will submit a single file, `vec.cpp` that includes the new class methods constructed as part of this lab.

sendlab.102.labsection# lab# vector.h