8CS112R

Normal Functions and Objects (Classes) – May pass in an object to a function any descendant will work if:

1. Passed by reference
2. The critical method is virtual

void printrounded (Shape &a)

{

cout << “the rounded area is “ << a.return\_rounded\_area() << endl;

}

main()

{

Circle cir;

Cylinder cly;

…

printrounded(cir);

printrounded(cyl);

}

Inline functions are designed to be very small, and to be used many times. The code is placed right where it is invoked. They should be very small.

Ex:

inline float

cube(float x)

{

return x \* x \* x;

}

Instead type as this:

inline float cube(float x) {return x \* x \* x;}

special if/else form (only to be used with inline functions)

test ? true action : false action

inline float max(float a, float b)

{

return a > b ? a : b;

}

cin >> \_\_ - skips white spaces

char ch1, ch2, ch3;

cin >> ch1 >> ch2 >> ch3;

to get all char input (even white spaces):

ch1 = cin.get();

ch = cin.get();

while(ch != ‘\n’)

{

cout << ch << endl;

ch = cin.get();

}

cin >> a;

cin.get();

ch=cin.get();

while(ch != ‘\n’)

while((ch=cin.get())

determine if an integer is a perfect number.

All the factors except itself add up to the number

Ex: 6: 1, 2, 3, ~~6~~

NOT

8: 1, 2, 4, ~~8~~

Input a number

Start sum at 0

Start factor at 1

While factor < number

If remainder from number/factor is 0

Add factor to sum

If sum equals number

Display yes

Else

Display No

1. ~~Want a function~~
2. Display each factor
3. Keep all the factors, don’t display

And count how many factors

Needs an array

1. Want a function, needs to return
   1. Yes/no – bool
   2. Array
   3. Count 🡨 automatically by reference

Receives a number, array, count

Start sum at 0

Sdtart factor at 1

Set count to 0

While factor < number

If remainer from number/factor is 0

Add factor to sum

Store factor in array (at location count)

Add 1 to count

Add to factor

If sum equals numner

return yes

Else

return no