Daniel Avila

CPSC 121

Chapter 7 Questions || Page 493

#6-13, 17 , 22, 24-30, 35, 52(a,b), 53

1. An object is a(n) instance of a class.
2. Creating a class object is often called instantiating the class.
3. Once a class is declared, how many objects can be created from it?
   1. 1
   2. 2
   3. Many
4. An object’s data items are stored in its member variables.
5. The procedures, or functions, an object performs are called its methods.
6. Bundling together an objects data and procedures is called encapsulation.
7. An object’s members can be declared public or private.

A public member can be accessed by functions outside the class.

A private member can be accessed by a function that is a member of the same class.

1. Normally a class’s member variables are declared to be private, and its member functions are declared to be public.
2. A class constructor is a member function with the same name as the class.
3. A destructor has the same name as the class but is preceded by a(n) tilde character.
4. A class may have more than one constructor, as long as each has a different parameters.
5. A class may only have one default constructor and one destructor.
6. In general, it is considered good practice to have member functions avoid doing input and output operations.
7. When a member function forms part of the interface through which a client program can use the class, the function must be public.
8. When a member function performs a task internal to the class and should not be called by a client program, the function should be made private.
9. True or False: A class object can be passed to a function but cannot be returned by a function. False. It can be passed and returned to a function.
10. True or False: C++ class objects are always passed to functions by reference. False. C++ class objects can be passed by value or reference.
11. By default, are the members of a structure public or private? Public
12. A)

class Circle: <- No colon needed here

{

private <- Colon needed here

double centerX;

double centerY;

double radius;

public <- Colon needed here

setCenter(double, double); <- No return type function ex. void

setRadius(double); <- No return type function ex. double

}

B)

Class Moon;<-No semi-colon needed

{

Private; <- “Private” should be lowercase and a colon is needed, not a semi-colon

double earthWeight;

double moonWeight;

Public; <- “Public” should be lowercase and a colon is needed, not a semi-colon

moonWeight(double ew);//Constructor <-Constructor needs to be same name as class and no semi-colon is need when defining the function

{ earthWeight = ew; moonWeight = earthWeight / 6; }

double getMoonWeight(); <-No need for semi-colon when defining

{ return moonWeight; }

} <-Needed with a semi-colon

int main ()

{

double earth;

cout >> “What is your weight? “;

cin << earth;

moon lunar(earth);

cout << “On the moon you would weigh “

<<lunar.getMoonWeigth() << endl;

return 0;

}

1. A)

class DumbBell; <- No Semi-colon here

{

int weight; <-There should be “private” before initializing the variable.

public:

void setWeigth(int);

};

void setWeigth(int w) <- it should be “void DumbBell::setWeigth(int w);”

{ weight = w; }

int main()

{

DumbBell bar;

DumbBell.setWeight(200); <-it should be “bar.setWeight(220);” because it needs the object

cout << “The weight is “ << bar.weight << endl; <- Needs a “getWeight()” for accessing private

return 0;

}

B)

class Change

{

private:

int pennies;

int nickels;

int dimes;

int quarters;

Change() <-Needs to be in the public || Both are default constructors

{ pennies = nickels = dimes = quarters = 0; } <-Needs to be in the public ||default constructor

Change (int p = 100, int n = 50, d = 50, q = 25); <-Needs to be in the public || default constructor

};

void Change::Change(int p ,int n, d, q) <-Specified Data Type for d and q

{

pennies = p;

nickels = m;

dimes = d;

quarters = q;

}