CS 31 Introduction to CS Discussion 2H

Structures

Structures

- Arrays provide a way to group data on a principle of data type
- Structures provide a way to group data logically (various data types)

How To Define A Structure?

- Use of **struct:** a reserved keyword in C/C++
- It defines a complex data type that consists of other complex data types or basic data types
- The variables defined in the body of a struct are called **attributes** of the structure

Example

```
struct product {
  int weight;
  float price;
} ;
```

Declaration Examples

```
product apple;                  //declaration
product banana, melon;          //declaration
```

Accessing Structure Attributes

- To access a structure attribute, we use the . (dot) operator
- Examples:
  - apple.weight = 10;
  - apple.price = 20;
  - banana.weight
  - banana.price
  - melon.weight
  - melon.price

Array of structures

```
product items[100];
items[1].price = 15;
cout << “Item 1 has price “  << items[1].price;
```

Nested structures

```
struct manufacturer {
  string companyName;
  string country;
};
struct product {
  manufacturer producedBy;
  int price;
}
```

Cascading Attribute Access

product banana;

banana.producedBy.country = “Africa”;
banana.producedBy.companyName = “BananaUnited”;

Classes and Objects

Objects

Objects

- Consider My Car:

PROPERTIES

Make: Honda
Model: Prelude

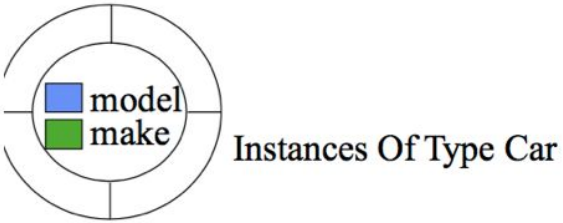
FUNCTIONALITY

play_music
toggle_left_blinker
honk

What are the main parts of an Object?

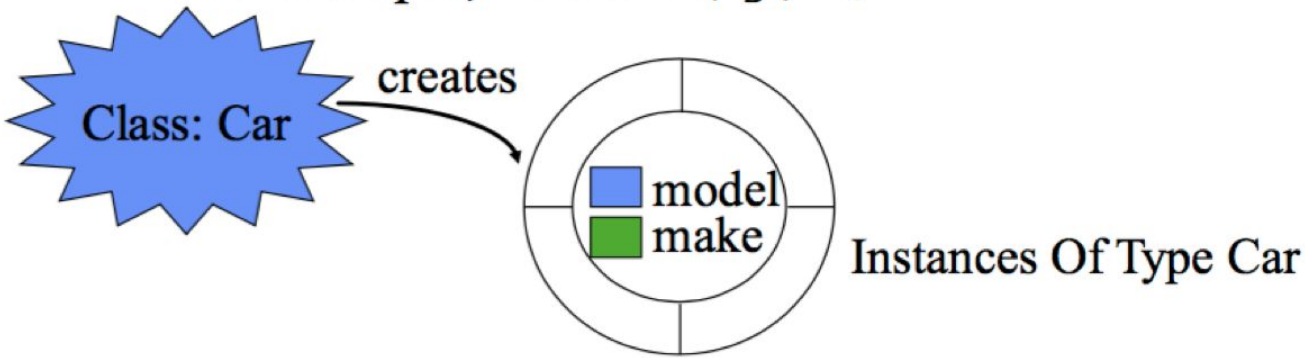
An Object Has...

- State Described Via Attributes
  - every car has a make and a model
- Behavior Described Via Methods
  - every car can honk its horn
- Identity Described Via Instances
  - from the sea of all Honda Preludes, I can identify the one that is mine



Classes

- Describe Similar Kinds Of Things
  - for example, consider the class of all `int`'s
- Programs Let Us Declare An Instance Of This Type
  - for example, `int i,j,k;`



Class example

```
class CRectangle {
    int width, height;
  public:
    CRectangle (int a,int b);
    int area () {return (width*height);}
};

CRectangle::CRectangle (int a, int b) {
    width = a;
    height = b;
}
```

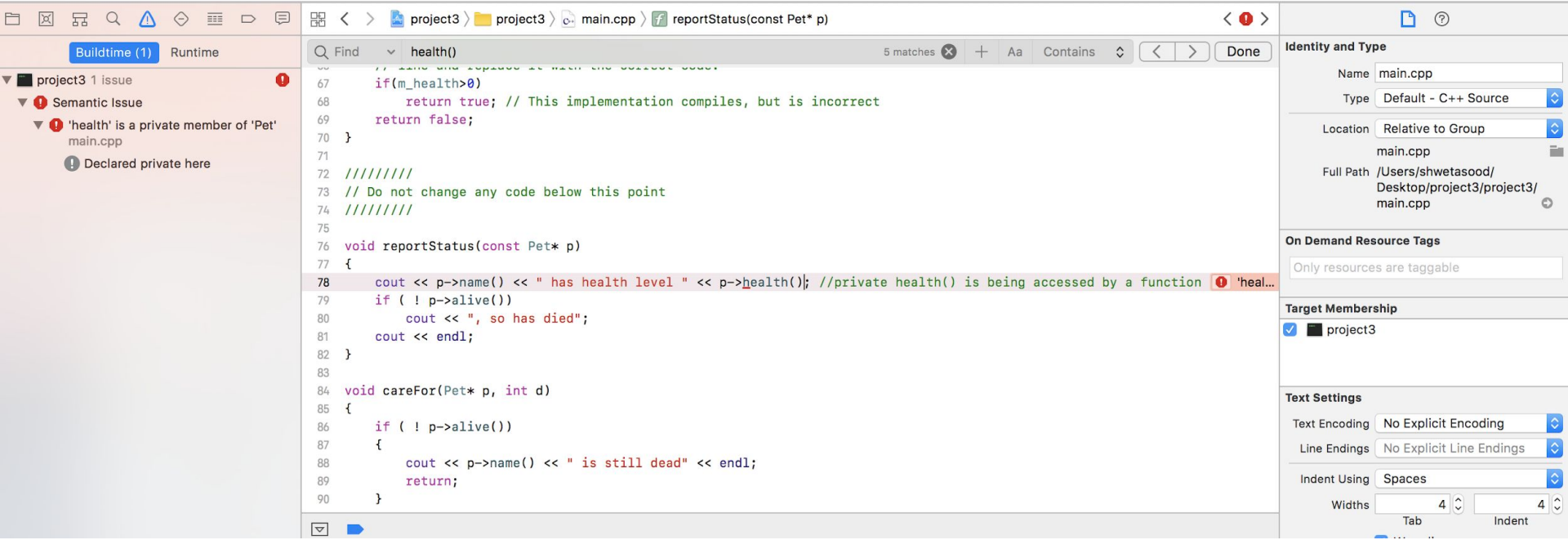
```
int main () {
  CRectangle rect (3,4);
  CRectangle rectb (5,6);
  cout << "rect area: " << rect.area() <<
endl;
  cout << "rectb area: " << rectb.area() <<
endl;
  return 0;
```

Time for Project 7 Warmup

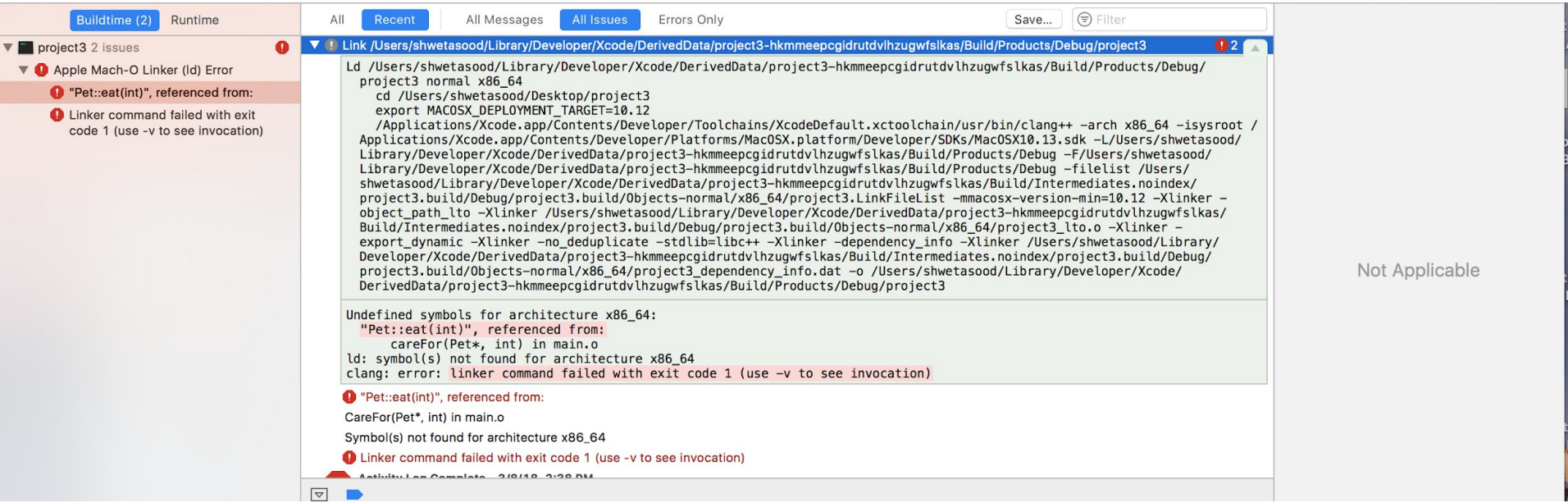
```
class Pet
{
  public:
    Pet(string nm, int initialHealth);
    void eat(int amt);
    void play();
    string name() const;
    int health() const;
    bool alive() const;
  private:
    string m_name;
    int m_health;
};

  // Initialize the state of the pet
Pet::Pet(string nm, int initialHealth)
{
    m_name = nm;
    m_health = initialHealth;
}

void Pet::eat(int amt)
{
      // TODO: Increase the pet's health by the amount
}

void Pet::play()
{
      // TODO: Decrease pet's health by 1 for the energy consumed
}

string Pet::name() const
{
      // TODO: Return the pet's name.  Delete the following line and
      // replace it with the correct code.
    return ""; // This implementation compiles, but is incorrect
}

int Pet::health() const
{
      // TODO: Return the pet's current health level.  Delete the
      // following line and replace it with the correct code.
    return 99; // This implementation compiles, but is incorrect
}

bool Pet::alive() const
{
      // TODO: Return whether pet is alive.  (A pet is alive if
      // its health is greater than zero.)  Delete the following
      // line and replace it with the correct code.
    return true; // This implementation compiles, but is incorrect
}
```

```
/////////
// Do not change any code below this point
/////////

void reportStatus(const Pet* p)
{
    cout << p->name() << " has health level " << p->health();
    if ( ! p->alive())
        cout << ", so has died";
    cout << endl;
}

void careFor(Pet* p, int d)
{
    if ( ! p->alive())
    {
        cout << p->name() << " is still dead" << endl;
        return;
    }

      // Every third day, you forget to feed your pet
    if (d % 3 == 0)
        cout << "You forgot to feed " << p->name() << endl;
    else
    {
        p->eat(1);  // Feed the pet one unit of food
        cout << "You fed " << p->name() << endl;
    }

    p->play();
    reportStatus(p);
}

int main()
{
    Pet* myPets[2];
    myPets[0] = new Pet("Fluffy", 2);
    myPets[1] = new Pet("Frisky", 4);
    for (int day = 1; day <= 9; day++)
    {
        cout << "======= Day " << day << endl;
        for (int k = 0; k < 2; k++)
            careFor(myPets[k], day);
    }
    cout << "=======" << endl;
    for (int k = 0; k < 2; k++)
    {
        if (myPets[k]->alive())
            cout << "Animal Control has come to rescue "
                 << myPets[k]->name() << endl;
        delete myPets[k];
    }
}
```

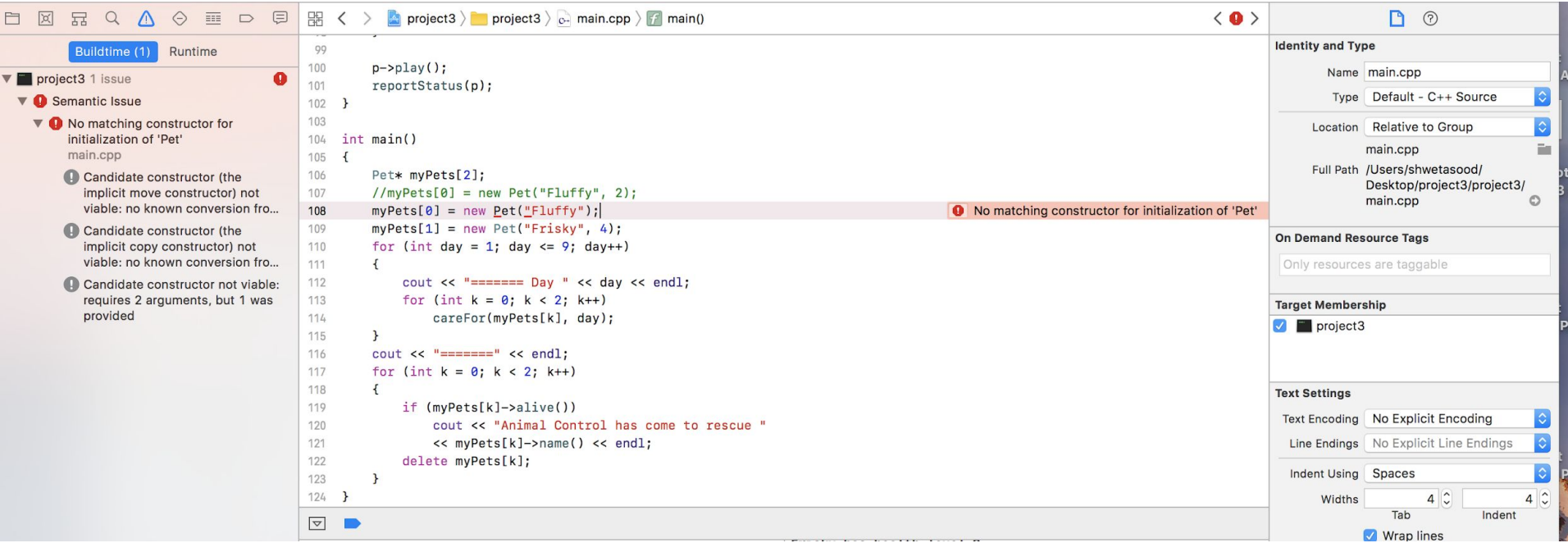
Try declaring the health member function `private` instead of `public`, and make sure you understand the resulting compilation error.


project3 › project3 › main.cpp › reportStatus(const Pet* p)
Buildtime (1)
Runtime
project3 1 issue
Semantic Issue
'health' is a private member of 'Pet'
main.cpp
Declared private here
Find
health()
5 matches
Aa
Contains
Done
67 if(m_health>0)
68 return true; // This implementation compiles, but is incorrect
69 return false;
70 }
72 /////////
73 // Do not change any code below this point
74 /////////
76 void reportStatus(const Pet* p)
77 {
78 cout << p->name() << " has health level " << p->health(); //private health() is being accessed by a function
79 if (! p->alive())
80 cout << ", so has died";
81 cout << endl;
82 }
84 void careFor(Pet* p, int d)
85 {
86 if (! p->alive())
87 {
88 cout << p->name() << " is still dead" << endl;
89 return;
90 }
Identity and Type
Name main.cpp
Type Default - C++ Source
Location Relative to Group
main.cpp
Full Path /Users/shwetasood/Desktop/project3/project3/main.cpp
On Demand Resource Tags
Only resources are taggable
Target Membership
project3
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 4
Tab
Indent

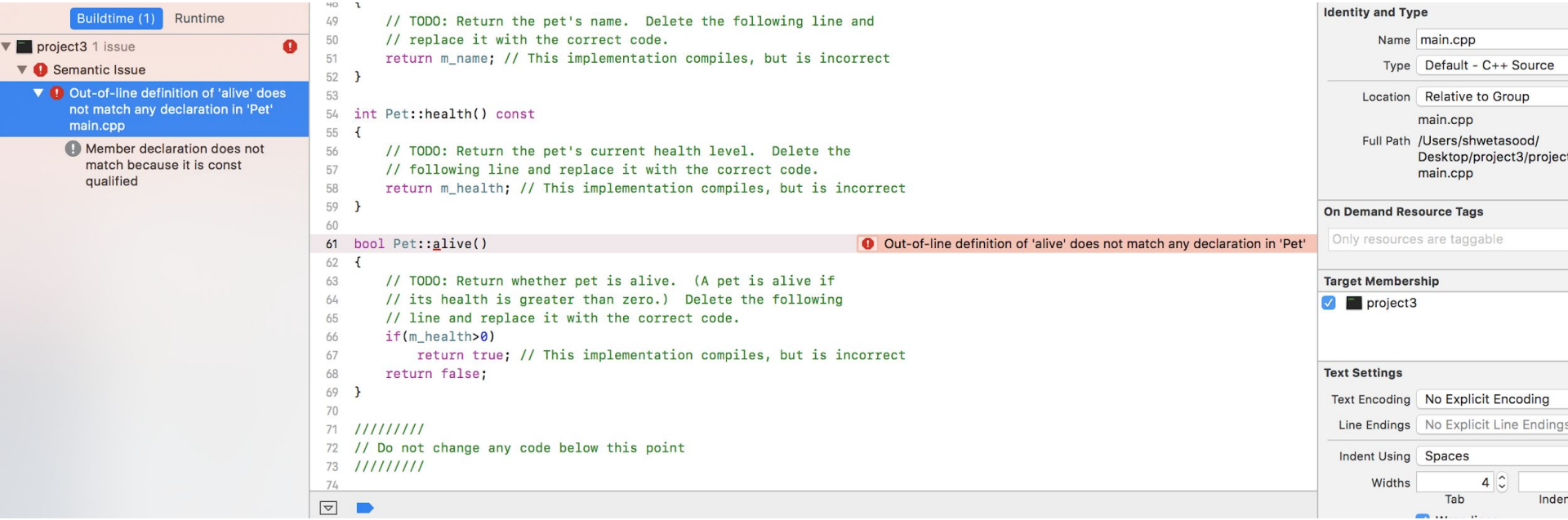
Comment out the entire implementation of `Pet::eat`, all the way from the `void Pet::eat(int amt)` to its close curly brace. Make sure you understand the resulting build error.


Buildtime (2)
Runtime
project3 2 issues
Apple Mach-O Linker (ld) Error
"Pet::eat(int)", referenced from:
Linker command failed with exit code 1 (use -v to see invocation)
All
Recent
All Messages
All Issues
Errors Only
Save...
Filter
Link /Users/shwetasood/Library/Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Products/Debug/project3
Ld /Users/shwetasood/Library/Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Products/Debug/
project3 normal x86_64
cd /Users/shwetasood/Desktop/project3
export MACOSX_DEPLOYMENT_TARGET=10.12
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang++ -arch x86_64 -isysroot /
Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.sdk -L/Users/shwetasood/
Library/Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Products/Debug -F/Users/shwetasood/
Library/Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Products/Debug -filelist /Users/
shwetasood/Library/Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Intermediates.noindex/
project3.build/Debug/project3.build/Objects-normal/x86_64/project3.LinkFileList -mmacosx-version-min=10.12 -Xlinker -
object_path_lto -Xlinker /Users/shwetasood/Library/Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/
Build/Intermediates.noindex/project3.build/Debug/project3.build/Objects-normal/x86_64/project3_lto.o -Xlinker -
export_dynamic -Xlinker -no_deduplicate -stdlib=libc++ -Xlinker -dependency_info -Xlinker /Users/shwetasood/Library/
Developer/Xcode/DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Intermediates.noindex/project3.build/Debug/
project3.build/Objects-normal/x86_64/project3_dependency_info.dat -o /Users/shwetasood/Library/Developer/Xcode/
DerivedData/project3-hkmmeepcgidrutdvlhzugwfslkas/Build/Products/Debug/project3
Undefined symbols for architecture x86_64:
"Pet::eat(int)", referenced from:
careFor(Pet*, int) in main.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
"Pet::eat(int)", referenced from:
CareFor(Pet*, int) in main.o
Symbol(s) not found for architecture x86_64
Linker command failed with exit code 1 (use -v to see invocation)
Not Applicable

In `main`, try replacing myPets[0] = new Pet("Fluffy", 2); with `myPets[0] = new Pet("Fluffy");` or `myPets[0] = new Pet;`, and make sure you understand the resulting compilation error.


Buildtime (1)
Runtime
project3 1 issue
Semantic Issue
No matching constructor for initialization of 'Pet'
main.cpp
Candidate constructor (the implicit move constructor) not viable: no known conversion fro...
Candidate constructor (the implicit copy constructor) not viable: no known conversion fro...
Candidate constructor not viable: requires 2 arguments, but 1 was provided
project3 〉 project3 〉 main.cpp 〉 main()
p->play();
reportStatus(p);
}
int main()
{
Pet* myPets[2];
//myPets[0] = new Pet("Fluffy", 2);
myPets[0] = new Pet("Fluffy");
No matching constructor for initialization of 'Pet'
myPets[1] = new Pet("Frisky", 4);
for (int day = 1; day <= 9; day++)
{
cout << "======= Day " << day << endl;
for (int k = 0; k < 2; k++)
careFor(myPets[k], day);
}
cout << "=======" << endl;
for (int k = 0; k < 2; k++)
{
if (myPets[k]->alive())
cout << "Animal Control has come to rescue "
<< myPets[k]->name() << endl;
delete myPets[k];
}
}
Identity and Type
Name main.cpp
Type Default - C++ Source
Location Relative to Group
main.cpp
Full Path /Users/shwetasood/Desktop/project3/project3/main.cpp
On Demand Resource Tags
Only resources are taggable
Target Membership
project3
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines

Try removing the `const` from the *implementation*, but not the *declaration* of the `alive` member function. Notice the compilation error.


Buildtime (1)
Runtime
project3 1 issue
Semantic Issue
Out-of-line definition of 'alive' does not match any declaration in 'Pet'
main.cpp
Member declaration does not match because it is const qualified
// TODO: Return the pet's name. Delete the following line and
// replace it with the correct code.
return m_name; // This implementation compiles, but is incorrect
}
int Pet::health() const
{
// TODO: Return the pet's current health level. Delete the
// following line and replace it with the correct code.
return m_health; // This implementation compiles, but is incorrect
}
bool Pet::alive()
Out-of-line definition of 'alive' does not match any declaration in 'Pet'
{
// TODO: Return whether pet is alive. (A pet is alive if
// its health is greater than zero.) Delete the following
// line and replace it with the correct code.
if(m_health>0)
return true; // This implementation compiles, but is incorrect
return false;
}
/////////
// Do not change any code below this point
/////////
Identity and Type
Name
main.cpp
Type
Default - C++ Source
Location
Relative to Group
main.cpp
Full Path
/Users/shwetasood/
Desktop/project3/project
main.cpp
On Demand Resource Tags
Only resources are taggable
Target Membership
project3
Text Settings
Text Encoding
No Explicit Encoding
Line Endings
No Explicit Line Endings
Indent Using
Spaces
Widths
4
Tab

Try removing the `const` from both the declaration and the implementation of the `alive` member function. Make sure you understand why the use of that function *doesn't* compile `in reportStatus`, but *does* compile in `careFor` and `main`.


project3 1 issue
Semantic Issue
Member function 'alive' not viable: 'this' argument has type 'const Pet', but function is not marked const
main.cpp
'alive' declared here
}
////////
// Do not change any code below this point
////////
void reportStatus(const Pet* p)
{
cout << p->name() << " has health level " << p->health(); //private health() is being accessed by a function
if (! p->alive())
Member function 'alive' not viable: 'this' argument has type 'const Pet', but function is not marked const
cout << ", so has died";
cout << endl;
}
void careFor(Pet* p, int d)
{
if (! p->alive())
{
cout << p->name() << " is still dead" << endl;
return;
}
// Every third day, you forget to feed your pet
if (d % 3 == 0)
Name main.cpp
Type Default - C++ Source
Location Relative to Group
main.cpp
Full Path /Users/shwetasood/Desktop/project3/project3/main.cpp
On Demand Resource Tags
Only resources are taggable
Target Membership
project3
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 4
Tab Indent

Time for practising Worksheet

Q1 a. What’s the main difference between declaring a type with the keyword struct and declaring it with the keyword class?

Q1 a. What’s the main difference between declaring a type with the keyword struct and declaring it with the keyword class?

struct has everything public by default if you don't specify otherwise, whereas class has everything private by default.

Q1 b. Why should you not allow data members to be public?

Q1 b. Why should you not allow data members to be public?

You do not want other users to be able to directly manipulate your object's data members (e.g., because they could set them to invalid values). By making data members private and providing a public interface that they must use, your implementation of that interface can control what values data members are set to.

Q1 c. What is the purpose of having private member functions in a class? Can you give some examples of when they would be used?

Q1 c. What is the purpose of having private member functions in a class? Can you give some examples of when they would be used?

Private functions are useful for placing code that the user does not need to be aware of but will be helpful for the implementations of your member functions. One example is a helper function with code that is common to two or more member functions.

Q1 d. (True/False) A class may have more than one constructor

Q1 d. (True/False) A class may have more than one constructor

True. You can overload constructors (as long as they differ in the number or type of arguments).

```
// overloading class constructors
#include <iostream>
using namespace std;

class Rectangle {
    int width, height;
  public:
    Rectangle ();
    Rectangle (int,int);
    int area (void) {return (width*height);}
};

Rectangle::Rectangle () {
  width = 5;
  height = 5;
}

Rectangle::Rectangle (int a, int b) {
  width = a;
  height = b;
}

int main () {
  Rectangle rect (3,4);
  Rectangle rectb;
  cout << "rect area: " << rect.area() << endl;
  cout << "rectb area: " << rectb.area() << endl;
  return 0;
}
```

Q1 e. (True/False) A class may have more than one destructor

Q1 e. (True/False) A class may have more than one destructor

False. A destructor doesn't have parameters, so there can be only one. However you can have more than 1 constructor since you can overload the constructor which is not possible with Destructors. Also to add that destructor is used to terminate the instance of the class and release all resources which it is using. There is nothing optional when you are destroying the object. The instance will not exist when destructor will be called.

Q1 f. If you have an object pointed by a pointer, which operator is used with the pointer to access the object's members?

Q1 f. If you have an object pointed by a pointer, which operator is used with the pointer to access the object's members?

‘->’ operator.

What will be the output of this?

```
#include <iostream>
#include <cstring>
#include <string>
using namespace std;

struct Stuff {
        int x;
        string s;
};

int main() {
        Stuff s;
        s.x = 5;
        s.s = "CS31";
        Stuff* ptr = &s;
        cout << ptr->x << endl;
        cout << ptr->s << endl;
}
```

Q1 g. What happens if you forget to deallocate memory once you’re done with the object?

Q1 g. What happens if you forget to deallocate memory once you’re done with the object?

You will have a memory leak which is problematic because it can lead to the program being unable to allocate memory at some point later on.

Q2. Write a class Person that has two private data members:

- m_age (an int)
- m_catchphrase (a string).

The Person class should have a default constructor that initializes its member variables to reasonable values and a second constructor that initializes the member variables to the values of its parameters. In addition, Person should have three public member functions:

- getAge(), which returns the Person’s age
- haveBirthday(), which increments the Person’s age by 1
- speak(), which prints the Person’s catchphrase.

```
class Person {
 public:
  Person() {
    m_age = 0;
    m_catchphrase = "";
  }
  Person(int age, int catchphrase) {
    m_age = age;
    m_catchphrase = catchphrase;
  }
  int getAge() const {
    return m_age;
  }
  void haveBirthday() {
    m_age++;
  }
  void speak() const {
    cout << m_catchphrase << endl;
  }
 private:
  int m_age;
  string m_catchphrase;
};
```

Const member functions in C++

2.6

A function becomes const when const keyword is used in function's declaration. The idea of const functions is not allow them to modify the object on which they are called. It is recommended practice to make as many functions const as possible so that accidental changes to objects are avoided.

Following is a simple example of const function.

```
#include<iostream>
using namespace std;

class Test {
    int value;
public:
    Test(int v = 0) {value = v;}

    // We get compiler error if we add a line like "value = 100;"
    // in this function.
    int getValue() const {return value;}
};

int main() {
    Test t(20);
    cout<<t.getValue();
    return 0;
}
```

Run on IDE

Q3. A line in Euclidean space can be represented by two parameters, m and b from its slope-intercept equation y = mx + b.

Here m represents the slope of the line and b represents the line's y-intercept. Write a class that represents a line.

Your class must have a simple constructor that initializes the line's m and b.

Next, define a member function with the following prototype: double intersection(Line line2);

This function must compute the x-coordinate where this line and another line (line2) intersect.

```
class Line {
public:
  Line(double m, double b) {
    m_m = m;
    m_b = b;
  }

  double m() const {
    return m_m;
  }

  double b() const {
    return m_b;
  }
  double intersection(Line line2) {
    if (m_m == line2.m()) {
      // same slope! SO the lines either are coincident or parallel
      // spec doesn't specify what we should do here, so return
      // whatever; in the real world we may want to throw an exception
      // (which aren't discussed in CS 31)
       return 0;
    }
    return (line2.b() - m_b)/(m_m - line2.m());
  }
private:
  double m_m;
  double m_b;
};
```

Bonus: There are two or three ways in which this problem specification is incomplete; they are not related to C++, but to the problem domain. What are they?

Bonus: There are two or three ways in which this problem specification is incomplete; they are not related to C++, but to the problem domain. What are they?

As mentioned in the comments above, the spec does not tell us what we should return if the two lines are coincident or parallel. Also, vertical lines cannot be exactly defined using the framework we have (e.g. x=3), although they can be approximated using a line with a large m.

Q4. Write a program that repeatedly reads an age and a catchphrase from the user and uses them to dynamically allocate a Person object, before calling the Person’s speak() function and then deallocating the Person object.

```cpp
#include <iostream>
#include <string>

using namespace std;

class Person {
public:
 Person() {
 m_age = 0;
 m_catchphrase = "";
 }
 Person(int age, string catchphrase) {
 m_age = age;
 m_catchphrase = catchphrase;
 }
 int getAge() const {
 return m_age;
 }
 void haveBirthday() {
 m_age++;
 }
 void speak() const {
 cout << m_catchphrase << endl;
 }
private:
 int m_age;
 string m_catchphrase;
};
int main() {
  int i=1;
  while(i!=0)
  {
    int age;
    string catchphrase;
    cout<<"Enter age: ";
    cin>>age;
    cin.ignore(10000,'\n');
    cout<<"Enter catchphrase: ";
    getline(cin, catchphrase);
    Person *p = new Person(age, catchphrase);
    p->speak();
    cout<<"Continue ?";
    cin>>i;
    delete p;
  }
}
```

```
const int NAME_LEN = 100;

class Cat {
      int m_age;
      char m_name[NAME_LEN];
      string m_type;
      Cat(int age, const char name[], string type) {
            m_age = age;
            m_name = name;
            type = type;
      }
   public:
      void introduce() {
            cout << "Hi! I am a " + type + " cat" << endl;
      }
};
```

```
struct Sheep {
      string m_name;
      int m_age;
      Sheep(int age) {
            m_age = age;
      }
      void introduce() {
            cout << "Hi! I am " + m_name + " the sheep" << endl;
      }
}

int main() {
      Cat* schrodinger = new Cat(5, "Schrodinger's cat",
"Korat");
      schrodinger->introduce();
      cout << schrodinger->m_age << endl;
      Sheep dolly(6);
      dolly->introduce();

      delete schrodinger;
      delete dolly;
}
```

What will the program above successfully print once all the fixes have been made?

Q5. Write a class called Complex, which represents a complex number.
Complex should have a default constructor and the following constructor:
Complex(int real, int imaginary); // -3 + 8i would be represented as Complex(-3, 8)

Additionally, the class should contain two functions: sum and print.
Sum should add two complex numbers. Print should print which complex number the object represents.
You may declare any private or public member variables or getters/setters you deem necessary.

```
class Complex {
    int m_real;
    int m_imaginary;
public:
    Complex() {}
    Complex(int real, int imaginary) {
        m_real = real;
        m_imaginary = imaginary;


    }
    void print() {
        cout << m_real << "+" << m_imaginary << "i" << endl;
    }
    void sum(Complex c1, Complex c2) {
        m_real = c1.m_real + c2.m_real;
        m_imaginary = c1.m_imaginary + c2.m_imaginary;
    }
};
```

```
int main() {
(1)   Complex c1(5, 6);
(2)   Complex c2(-2, 4);
(3)   Complex* c3 = new Complex();

(4)   c1.print();
(5)   c2.print();
(6)   cout << "The sum of the two complex numbers is:" << endl;
(7)   c3->sum(c1, c2);
(8)   c3->print();
(9)   delete c3;
}

// The output of the main program:
5+6i
-2+4i
The sum of the two complex numbers is:
3+10i
```

What would happen if swapped the order of (8) and (9)? How would it change the output?

After deleting the object pointed to by c3, an attempt to follow the pointer c3 is undefined behavior. The program might crash, print weird values (perhaps because the memory used by the deleted object was overwritten with some bookkeeping information the storage manager uses), print 3+10i (if the memory used was not overwritten), or do something else.

8. Suppose you have a struct defined as follows:

```
struct Array {
      int* vals;
      int len;
};
```

Within Array, `vals` is a pointer to an array of ints (that is *not* dynamically allocated). The field `len` describes the length of this array.

Design a function with the following header:

```
int findArrayWithMax(Array arr1, Array arr2, Array arr3);
```

Given three Arrays *arr1*, *arr2*, and *arr3*, this function should return the number of the Array that contains the maximum value of the three Arrays. If the Array with the maximum value is *arr1*, it should return 1 (2 for *arr2* and 3 for *arr3*).

```
int a[5] = {3, 4, 5, 6, 1};
int b[2] = {1000, -1};
int c[9] = {23, 2, 1, 4, 65, 42, 10, -20, 7};
Array arr1 = { a, 5 };
Array arr2 = { b, 2 };
Array arr3 = { c, 9 };
int max = findArrayWithMax(arr1, arr2, arr3); // max = 2
```

```
void updateMax(Array arr, int& max, int& maxNum, bool&
numFound,
               int arrNum) {
  for (int x = 0; x < arr.len; x++) {
    if (!numFound || arr.vals[x] > maxNum) {
      max = arrNum;
      maxNum = arr.vals[x];
      numFound = true;
    }
  }
}

int findArrayWithMax(Array arr1, Array arr2, Array arr3) {
  int max = 0;
  int maxNum;
  bool numFound = false;

  updateMax(arr1, max, maxNum, numFound, 1);
  updateMax(arr2, max, maxNum, numFound, 2);
  updateMax(arr3, max, maxNum, numFound, 3);
  return max;
}
```

Thank You!