5

58

DEPARTMENT OF COMPUTER SCIENCES AND SOFTWARE ENGINEERING
AUBURN UNIVERSITY

COMP 2710: Software Construction
Fall 2014

Quiz 2

Nov. 18, 2014
12:30pm – 1:10pm

Name: Kristi Reckart

This quiz contains 8 questions; make sure your copy has them all. This is a closed-book quiz. Write all your answers in these question sheets.

1. Consider the following code fragment.

```
class Pair
{
public:
    Pair();
    Pair(int firstVal, int secondVal);
    void setFirst(int newVal);
    void setSecond(int newVal);
    int getFirst() const;
    int getSecond() const;
private:
    int first; int second;
};
```

   a) The main problem with the above codes is that it allows only creation of pairs of integer values. Re-write the above codes that will allow you to create objects that are Pairs of different types of values, such as double and string values.

```
class Pair<T>
{
public:
    Pair<T>();
    Pair<T>(T firstVal, T secondVal);
    void setFirst(T newVal);
    void setSecond(T newVal);
    T getFirst() const;
    T getSecond() const;
private:
    T first, second;
};
```

5

12

Pair <double> dPair (2.5, 3.5)

my answer still got credit but this was what he wanted, have "double" replace the T's in constructor => T's here become double

b) For the codes that you have re-written above, write the codes that will create a Pair object of double values?

~~Pair<T> (T firstVal, T secondVal)~~ ✓

setFirst (T newVal);

setSecond (T newVal);

c) Write the codes that will print out the sum of the two values of the Pair object that you have created in (b) above.

my answer still correct, but this was what he wanted

```
T getFirst() const;
T getSecond() const;
double num = dPair.getFirst() + dPair.getSecond();
cout << num;
```

2. Consider writing into a file using a write (or insertion) operator.

```
#include <fstream>
using namespace std;
ofstream outStream;
outStream.open("outfile.txt");
outStream << "Test Data" << endl;
```

a) Suppose you are using the gdb debugger and you are executing one line at a time. After executing the line outStream << "Test Data" << endl, you check the output file outfile.txt and the string "Test Data" is not in the output file outfile.txt. Explain why the string "Test Data" may not be in the output file outfile.txt *and* where is the string "Test Data" at this point?

the string is in the buffer between the code & file; it has not been written b/c the data in buffer has not been flushed by using either outStream.flush() or outstream.close()

12

b) Write the code that will force the string "Test Data" to be written out into the output file outfile.txt, assuming that only local file system is used and no NFS (Network File System) is used.

outStream.flush()

c) If NFS is used for the output file outfile.txt, explain why the code that you write in (b) will not ensure that the string "Test Data" will be written into the output file outfile.txt.

b/c changes to a file in a NFS will not be saved and appear until the file is closed

for everyone to see

d) Write the codes that will ensure that the string "Test Data" to be written out into the output file outfile.txt, even if NFS is used.

outStream.close()

3. Consider the following definitions of the Employee and HourlyEmployee classes:

```
class Employee
{
 public:
    virtual void retrieve_employeeInfo()   { cout << "Retrieve employee Info" << endl; }
    virtual void calculate_grossPay()   { cout << "Employee Gross Pay" << endl; }
    virtual void withhold_tax()  { cout << "Employee Tax Withheld" << endl; }
    virtual void calculate_netPay()  { cout << "Employee Net Pay " << endl; }
        void process_paycheck() { retrieve_employeeInfo(); calculate_grossPay();
                withhold_tax(); calculate_netPay();  }
 private:
    string name;
 };

class HourlyEmployee : public Employee
{
 public:
    void retrieve_employeeInfo ()   { cout << "Hourly Employee Info" << endl; }
    void calculate_grossPay() { cout << "Gross Pay using Hourly Employee Method" << endl; }
    void withhold_tax() { cout << "Tax Withheld using Hourly Employee Method " << endl; }
    void calculate_netPay() { cout << "Net Pay using Hourly Employee Method " << endl; }
 private:
    double rate;
 };

main()
{
     HourlyEmployee he;

     he.process_paycheck();  // Use hourly employee's method for retrieving
                             // employee info, calculate gross pay, tax withheld
                             // and the net pay.
}
```

a) Will the above codes correctly perform the intended operations in the main function? If so, explain why it is correct. If not, explain why it is incorrect.

No, there is no HourlyEmployee constructor which calls upon the Employee constructor, and there is no process_paycheck() w/in HourlyEmployee to override the Employee's method & use it's own methods for retrieving data.

can correct by making functions virtual; this results in process_paycheck() calling the functions of HourlyEmployee

4

b) If the above codes are correct, write the outputs that are written by the codes. If they are incorrect, re-write the above codes so that the codes in the main function will correctly perform their intended operations and write the outputs that are written by the codes that you have re-written.

```
main () {
    HourlyEmployee
        he = HourlyEmployee();
    he.process_
        payCheck();
}

class Employee {
    Employee() {};
    ...
};

class HourlyEmployee {
    HourlyEmployee() : Employee() {};
    void process_paycheck() {retrieve_employeeInfo();
        calculate_grossPay(); withhold_tax();
        calculate_
        netPay() };
};
```

virtual functions

c) If the above codes are correct, explain the mechanism that makes them correct. If they are incorrect and you have re-written the codes in (b), explain the mechanism that makes the codes you provided operate correctly.

The new function w/in the HourlyE mployee class overrides the Employee class' process_payCheck() method to create it's own process_payCheck() method in the HourlyEmployee class that calls the other functions w/in HourlyEmployee that have already been overridden.

3

virtual function → late binding
virtual fun. table → wait to call the correct fun. when fun. is used

4. Suppose a random access file consists of records of the `StudentRec` objects, where `StudentRec` is a user-defined class. The random access file is opened with the stream object `ioStream`.

a) What is the class of the `ioStream` object?

fstream ioStream;

The constructor ensures that A space for "name" of the Employee class will be created for HourlyEmployee class

4

586-1 = 585

b) Write the code fragment to position the get-pointer at the 586th `StudentRec` object in the file.

```
iostream.seekg(585 * sizeof(StudentRec));
iostream>>SR;
```

for reading

size of the StudentRec class / object

c) Write the code fragment to position the put-pointer so that some subsequent code can write a `StudentRec` objects at the fifth `StudentRec` object position in the file.

every StudentRec object is of the same class, so same size

SR will be written

```
iostream.seekp(4 * sizeof(StudentRec));
iostream << SR;
```

5. Consider the following member function prototype in a Money class:

```
const Money operator +(const Money& amount) const;   // version 1
```

a) What is the meaning of const at the *beginning* of the function prototype?

the value on the left side of the + operator can't be changed

4

the Money object can't be modified

```
sum = cost + tax;
    = cost.+(tax);
```

sum Money anonymous obj.

b) What is the meaning of const at the *end* of the function prototype?

returns a value that can't be changed

4

the cost parameter can't be modified

c) Compare the above with a different overload of the operator + as follows:

const Money operator +(Money amount) const; // version 2

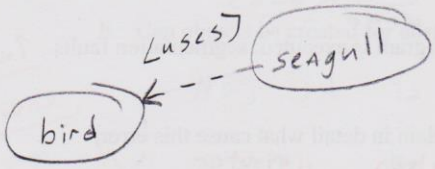
Explain why one version is more efficient than the other.

this is allowed to be changed which shouldn't happen; the function should get data not change it.

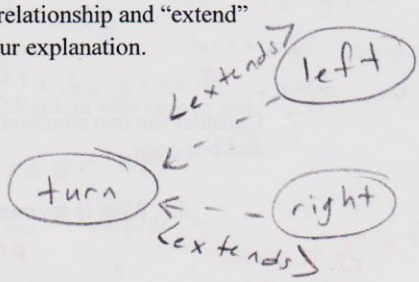
call by reference (version 1) more efficient; only pass the address (very few bytes)

6. Consider use case relationships.

a) What is the difference between "uses/include" relationship and "extend" relationship? Use diagrams and examples in your explanation.



seagull uses exact duplicates of methods from the bird class



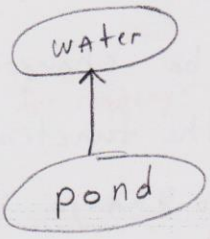
left and right override functions of the turn class to be tailored to the specifically different classes

uses/includes → uses A more common behavior (use cas)

extend → add actions

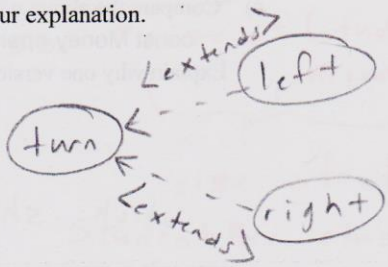
4

b) What is the difference between "generalization" relationship and "extend" relationship? Use diagrams and examples in your explanation.



4

water is a generalization of pond; has more methods; pond use same methods that can be made very specific to suit pond

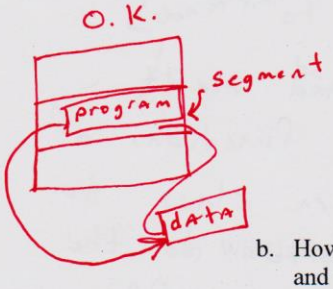


look @ (A)

generalization → a use case is on special type of another

7. Consider the two common errors when a program is executed: segmentation faults and bus error.

a. What is segmentation fault? Explain in detail what cause this error.



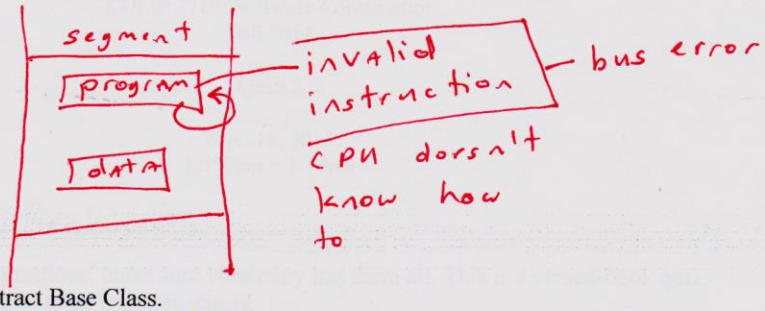
problem when pointer accesses something outside the segment given to program by operating system

b. How do you find *which line of the code* the segmentation fault occurred and *what variable* caused it?

gdb
> backtrace — OR — where
> print pointer — if ∅ bad pointer
> print *pointer

8

c. What is bus error? Explain in detail what cause bus error.



8. Consider Abstract Base Class.

a. What are abstract base classes?

A class that has all functions that are pure virtual functions

An outline that can be used for other classes to make objects w/out having to retype code

b. Can objects be created for abstract base classes? Explain why or why not.

No, it is a blueprint that must be able to apply to other classes it is made up of virtual functions that are undefined

c. If a class is derived from an abstract base class, under what conditions will the derived class also be an abstract class? Explain.

when the derived class inherits the pure virtual function & doesn't define them