CS 253 Midterm 1 – Fall 2015

Name _____ ANSWER KEY _____
EID ____________________________

Instructions:

This exam is closed book, closed notes, etc. You may not use external sources of information of any kind while taking this exam.

All questions are multiple-choice and you MUST mark your answers on the Scantron sheet you are being provided. Both this exam and the Scantron sheet will be collected at the end of the exam. Please USE PENCIL on the Scantron sheets.

Generally, the exam pages will be archived and not reviewed; all grading will be based upon your Scantron sheets. That said, it is recommended that you mark your answers also on this exam as a backup should any questions arise regarding the automated grading process.

As has been discussed in class and presented on the class website, this exam is entirely based upon two object classes already distributed to every student in the course and as defined in the C++ code from the files Animals.h and Animals.cpp. Both of these files are reproduced as part of this exam on the pages following of the actual exam questions.

In addition, new code in the form of a main.cpp file is being provided with this exam. You will see this code on the last two pages of the exam. The questions you are being asked relate to the six examples in main.cpp and how they will behave given the code you have previously studied in Animals.h and Animals.cpp.

There are a total of 20 questions and each question is worth 5 points.

Good Luck.
**Question 1:**

What best describes the behavior of the program you are being given when compiled as:

```bash
g++ -I. -std=c++11 main.cpp Animals.cpp -o runit
```

and then invoked with the following at the command line:

```bash
> runit 8
```

Your choices: *(the correct answer is C)*

a) It runs example 1 by default as defined in the function `example_1()`  
b) It prints only 2 messages: “Welcome to ...” and “Say Goodbye ...”  
c) It prints “runit should be called with a single numeric argument between 1 and 6”  
d) It terminates abnormally with a segmentation fault

**Question 2:**

In this and all subsequent questions assume the program has been compiled with the command described in Question 1. Now, when this command is run:

```bash
> runit 1
```

How many Mice are created? *(the correct answer is B)*

a) Zero  
b) One  
c) Two  
d) Three  
e) More than three

**Question 3:** When this command is run:

```bash
> runit 1
```

then the line of code in example 1

```cpp
e1.walkTheLine(6);
```

will generate which of the following as terminal output? *(The correct answer is C)*

a) “Hey There from Dumbo and Timothy” repeated six times  
b) “Hey There from Timothy and Dumbo” repeated six times  
c) “Hey There from Dumbo and Timothy”  
d) “Hey There from Timothy and Dumbo”  
e) None of the above
**Question 4:** When this command is run:

```
> runit 2
```

then the line of code in example 2

```java
e1.walkTheLine(6);
```

will generate which of the following as terminal output? *(The correct answer is A)*

a) “Hey There from Dumbo and Timothy” repeated six times  
b) “Hey There from Timothy and Dumbo” repeated six times  
c) “Hey There from Dumbo and Timothy”  
d) “Hey There from Timothy and Dumbo”

**Question 5:** When this command is run:

```
> runit 3
```

then what best describes the number of animals created? *(The correct answer is C)*

a) One Mouse (Mice) and one elephant  
b) One Elephant and three Mice  
c) One Mouse (Mice) and three Elephants  
d) Three Mice and three Elephants  
e) Six Mice and three Elephants

**Question 6:** When this command is run:

```
> runit 3
```

then the Elephant member function `join` is being called three times. Which option below best describes how information about other elephants is being provided to this function. *(The correct answer is B)*

a) They are being passed by value  
b) They are being passed by reference  
c) They are globally available on the heap  
d) They are globally available on the stack  
e) They are being passed by name – a string
**Question 7:** When this command is run:

```
> runit 3
```

then the line of code

```java
e1.walkTheLine(6);
```

will result in which of the following being printed to the terminal output

(The correct answer is D)

- a) Three lines:
  
  Hey There from Catty and Timothy
  Hey There from Prissy and Timothy
  Hey There from Giddy and Timothy

- b) Three lines:
  
  Hey There from Catty and Timothy
  Hey There from Giddy and Timothy
  Hey There from Prissy and Timothy

- c) Six lines:
  
  Hey There from Catty and Timothy
  Hey There from Prissy and Timothy
  Hey There from Giddy and Timothy
  Hey There from Catty and Timothy
  Hey There from Prissy and Timothy
  Hey There from Giddy and Timothy

- d) Six lines:
  
  Hey There from Catty and Timothy
  Hey There from Giddy and Timothy
  Hey There from Prissy and Timothy
  Hey There from Catty and Timothy
  Hey There from Giddy and Timothy
  Hey There from Prissy and Timothy

- e) None of the above
**Question 8:** When this command is run:

```bash
>runit 4
```

then what best describes the number of animals created? *(The correct answer is D)*

a) One Mouse (Mice) and one elephant  
b) One Elephant and three Mice  
c) One Mouse (Mice) and three Elephants  
d) Three Mice and three Elephants  
e) Six Mice and three Elephants

**Question 9:** When this command is run:

```bash
>runit 4
```

then the line of code

```java
e1.walkTheLine(3);
```

will result in which of the following being printed to the terminal output *(The correct answer is B – see correction to labels as well)*

f) Three lines: *(Option A)*

   Hey There from Catty and Jerry  
   Hey There from Prissy and Itchy  
   Hey There from Giddy and Mickey


g) Three lines: *(Option B)*

   Hey There from Catty and Jerry  
   Hey There from Giddy and Mickey  
   Hey There from Prissy and Itchy

h) Three lines: *(Option C)*

   Hey There from Catty and Jerry  
   Hey There from Prissy and Mickey  
   Hey There from Giddy and Itchy

i) Three lines: *(Option D)*

   Hey There from Catty and Jerry  
   Hey There from Giddy and Itchy  
   Hey There from Prissy and Mickey

j) None of the above *(Option E)*
Question 10: When this command is run:
>runit 4

the last three lines printed to the terminal before the program terminates are:

(The correct answer is C)

Option A:

..... Say goodbye to the Elephants.
-- Deconstructing Elephant Catty
-- Deconstructing Mouse Jerry

Option B:

..... Say goodbye to the Elephants.
-- Deconstructing Mouse Jerry
-- Deconstructing Elephant Catty

Option C:

-- Deconstructing Elephant Catty
-- Deconstructing Mouse Jerry
..... Say goodbye to the Elephants.

Option D:

-- Deconstructing Elephant Giddy
-- Deconstructing Elephant Catty
..... Say goodbye to the Elephants.

Option E: None of the above
**Question 11:** When this command is run:

```bash
>runit 5
```

then the line of code

```java
e1.walkTheLine(6);
```

will result in which of the following being printed to the terminal output

(The correct answer is D)

**Option A:**

- Hey There from Catty and Jerry
- Hey There from Giddy and Mickey
- Hey There from Prissy and Itchy

**Option B:**

- Hey There from Catty and Jerry
- Hey There from Giddy and Mickey
- Hey There from Prissy and Itchy
- Hey There from Catty and Jerry
- Hey There from Giddy and Mickey
- Hey There from Prissy and Itchy

**Option C:**

- Hey There from Catty and Jerry
- Hey There from Prissy and Itchy

**Option D:**

- Hey There from Catty and Jerry
- Hey There from Prissy and Itchy
- Hey There from Catty and Jerry
- Hey There from Prissy and Itchy
- Hey There from Catty and Jerry
- Hey There from Prissy and Itchy

**Option E:** None of the above
**Question 12:** When this command is run:

>`runit 5`

and if the line of code reading

```java
   e1.walkTheLine(6);
```

is replaced with a line reading

```java
   e2.walkTheLine(6);
```

which of the following will happen? *(The correct answer is C)*

**Option A:**

There will be a run-time segmentation fault

**Option B:**

The following will be printed to the terminal

```
Hey There from Catty and Jerry
Hey There from Prissy and Itchy
```

**Option C:**

The following will be printed to the terminal

```
Hey There from Giddy and Mickey
Hey There from Prissy and Itchy
Hey There from Catty and Jerry
Hey There from Prissy and Itchy
Hey There from Catty and Jerry
Hey There from Prissy and Itchy
```

**Option D:**

The following will be printed to the terminal

```
Hey There from Catty and Jerry
Hey There from Prissy and Itchy
Hey There from Catty and Jerry
Hey There from Prissy and Itchy
Hey There from Catty and Jerry
Hey There from Prissy and Itchy
```

**Option E:** None of the above
**Question 13:** When this command is run:

> `runit 6`

then what best describes the resulting memory usage. *(The correct answer is C)*

- Option A: Three Mice and three Elephants on the stack
- Option B: Three Mice and three Elephants on the heap
- Option C: Three Mice on the heap and three Elephants on the stack
- Option D: Three Mice on the stack and three Elephants on the heap
- Option E: None of the above

**Question 14:** Now consider all six numbered examples and memory utilization and pick the best description relative to memory leaks. *(The correct answer is D)*

- Option A: There are no memory leaks in any of the six examples.
- Option B: Memory is leaking in all of the examples because objects are being constructed within the scope of `main()`.
- Option C: There are memory leaks in examples four, five and six.
- Option D: There are memory leaks in example six.

**Question 15:** Again consider the six numbered examples. Which of them use the default constructor for objects of class Mice? *(The correct answer is B)*

- Option A: None
- Option B: Examples 1, 2 and 3.
- Option C: Examples 4, 5, and 6.
- Option D: All of the examples
- Option E: There is no user defined default constructor
**Question 16:** Return to example 3 and consider this change.

The line

```plaintext
e3.join(& e2, & e1);
```

is replaced with

```plaintext
e3.join(& e2, NULL);
```

now pick the best description of the output generated by the line

```plaintext
e1.walkTheLine(6);  (The correct answer is B)
```

a) Three lines:
- Hey There from Catty and Timothy
- Hey There from Prissy and Timothy
- Hey There from Giddy and Timothy

b) Three lines:
- Hey There from Catty and Timothy
- Hey There from Giddy and Timothy
- Hey There from Prissy and Timothy

c) Six lines:
- Hey There from Catty and Timothy
- Hey There from Prissy and Timothy
- Hey There from Giddy and Timothy
- Hey There from Catty and Timothy
- Hey There from Prissy and Timothy
- Hey There from Giddy and Timothy

d) Six lines:
- Hey There from Catty and Timothy
- Hey There from Giddy and Timothy
- Hey There from Prissy and Timothy
- Hey There from Catty and Timothy
- Hey There from Giddy and Timothy
- Hey There from Prissy and Timothy

e) None of the above
**Question 17:** Consider again example 1 and best describe what happens if the line

```cpp
delete & el;
```

is added as the last line to the function `example_1` *(The correct answer is B)*

- **Option A:** The elephant is deconstructed from the stack slightly faster than might otherwise be the case.
- **Option B:** There is in all likelihood a run-time error due to trying to free memory allocated on the stack.
- **Option C:** The code cannot and will not compile because the associated Mouse is not also deleted.
- **Option D:** The code will not compile because an object is being passed to delete instead of a reference to an object.
- **Option E:** There is no change at all to how the code will behave.

**Question 18:** What best describes the nesting relationship between one Elephant and one Mouse? *(The correct answer is C)*

- **Option A:** Inside every Elephant there is a unique Mouse instance very literally sharing the same memory used to store the Elephant.
- **Option B:** Inside every Elephant there is a default Mouse and sometimes another Mouse instance passed in at the time the Elephant is constructed.
- **Option C:** Inside every Elephant is a reference to one and only one Mouse.
- **Option D:** Inside every Elephant is the capability to assign zero or more friendly Mice.
- **Option E:** Elephants are afraid of Mice and Elephants never ever associate with Mice.

**Question 19:** How many explicit copy constructors are defined in the file `Animals.h` *(The correct answer is A)*

- **Option A:** Zero
- **Option B:** One
- **Option C:** Two
- **Option D:** Three
- **Option E:** Four
**Question 20:** One of these statements is a far better high-level description of the Animals code than the others. Pick the best one: *(The correct answer is D)*

Option A: This code implements singly-linked lists of Elephants with the possibility to nesting Elephants one within another within another.

Option B: This code implements doubly-linked lists of Elephants with the possibility to nesting Elephants one within another within another.

Option C: This code implements singly-linked lists of Elephants with the possibility of connecting Elephants one to the next.

Option D: This code implements doubly-linked lists of Elephants with the possibility of connecting Elephants one to the next to the next.

Option E: This code implements Elephant trees with Elephants including friends who are Mice.
The files follow:

```cpp
// Animals.h
#ifndef ANIMALS_H
#define ANIMALS_H

#include <iostream>
using namespace std;

class Mouse {
public:
    Mouse();
    Mouse(const string & n);
    ~Mouse();
    string getName();

protected:
    string name;
};

class Elephant {
public:
    Elephant(Mouse & m);
    Elephant(const string n, Mouse & m);
    ~Elephant();
    void setAhead(Elephant * e);
    void setBehind(Elephant * e);
    void join(Elephant * front, Elephant * back);
    Elephant * getAhead();
    Elephant * getBehind();
    void speak();
    Elephant * remove();
    int walkTheLine(int steps);

protected:
    string name;
    Mouse & eFriend;
    Elephant * ahead = NULL;
    Elephant * behind = NULL;
};

#endif // ANIMALS_H
```
// Animals.cpp
#include <Animals.h>

Mouse::Mouse() {
    name = "Timothy";
    cout << "++ Default Construct Mouse: " << name << endl;
}
Mouse::Mouse(const string & n) : name(n) {
    cout << "++ Named Mouse Construct: " << name << endl;
}
Mouse::~Mouse() {
    cout << "-- Deconstructing Mouse " << name << endl;
}
string Mouse::getName() {
    return name;
}

Elephant::Elephant(Mouse & m) : eFriend(m) {
    name = "Dumbo";
    ahead = NULL;
    behind = NULL;
    cout << "++ Default Construct Elephant: " << name << endl;
}
Elephant::Elephant(const string n, Mouse & m) : name(n), eFriend(m) {
    ahead = NULL;
    behind = NULL;
    cout << "++ Named Construct Elephant: " << name << endl;
}
Elephant::~Elephant() {
    cout << "-- Deconstructing Elephant " << name << endl;
}
void Elephant::speak() {
    cout << "Hey There from " << name << " and " << eFriend.getName() << endl;
}
void Elephant::setAhead(Elephant * e) {
    ahead = e;
}
void Elephant::setBehind(Elephant * e) {
    behind = e;
}
void Elephant::join(Elephant * front, Elephant * back) {
    ahead = front;
    behind = back;
}
Elephant * Elephant::getAhead() {
    return ahead;
}
Elephant * Elephant::getBehind() {
    return behind;
}
Elephant * Elephant::remove() {
    Elephant * me = this;
    Elephant * front = me->getAhead();
    Elephant * back = me->getBehind();
    front->setBehind(back);
    back->setAhead(front);
    return back;
}
int Elephant::walkTheLine(int steps) {
    Elephant * e = this;
    for (int i = 0; i < steps; i++) {
        e->speak();
        e = e->getBehind();
        if (e == NULL) return -1;
    }
    return 0;
}
```c
#include <Animals.h>
#include <sstream>
using std::istringstream;

int Usage(char* argv0)
{
    cout << argv0 << " should be called with a single numeric argument between 1 and 6" << endl;
    return -1;
}

void example_1()
{
    Mouse m = Mouse();
    Elephant e1 = Elephant(m);
    e1.walkTheLine(6);
    delete & e1;
}

void example_2()
{
    Mouse m = Mouse();
    Elephant e1 = Elephant(m);
    e1.setAhead(& e1);
    e1.setBehind(& e1);
    e1.walkTheLine(6);
}

void example_3()
{
    Mouse m = Mouse();
    Elephant e1 = Elephant("Catty", m);
    Elephant e2 = Elephant("Giddy", m);
    Elephant e3 = Elephant("Prissy", m);
    e1.join(& e3, & e2);
    e2.join(& e1, & e3);
    e3.join(& e2, & e1);
    e1.walkTheLine(6);
}

void example_4()
{
    Mouse a = Mouse("Jerry");
    Elephant e1 = Elephant("Catty", a);
    Mouse b = Mouse("Mickey");
    Elephant e2 = Elephant("Giddy", b);
    Mouse c = Mouse("Itchy");
    Elephant e3 = Elephant("Prissy", c);
    e1.join(& e3, & e2);
    e2.join(& e1, & e3);
    e3.join(& e2, & e1);
    e1.walkTheLine(3);
}
```
```cpp
void example_5() {
    Mouse a = Mouse("Jerry");
    Elephant e1 = Elephant("Catty", a);
    Mouse b = Mouse("Mickey");
    Elephant e2 = Elephant("Giddy", b);
    Mouse c = Mouse("Itchy");
    Elephant e3 = Elephant("Prissy", c);
    e1.join(& e3, & e2);
    e2.join(& e1, & e3);
    e3.join(& e2, & e1);
    e2.remove();
    e1.walkTheLine(6);
}

void example_6() {
    Elephant e1 = Elephant("Catty",  * new Mouse("Jerry"));
    Elephant e2 = Elephant("Giddy",  * new Mouse("Mickey"));
    Elephant e3 = Elephant("Prissy", * new Mouse("Itchy"));
    e1.join(& e3, & e2);
    e2.join(& e1, & e3);
    e3.join(& e2, & e1);
    e1.walkTheLine(3);
}

int main(int argc, char* argv[]) {
    if (argc != 2) return Usage(argv[0]);

    istringstream istr(argv[1]);
    int keyvalue;
    istr >> keyvalue;
    if ((istr.fail()) return Usage(argv[0]);
    if ((keyvalue < 0) || (keyvalue > 7)) return Usage(argv[0]);

    cout << "...... Welcome to Elephant Land." << endl;

    switch(keyvalue) {
    case 1 : { example_1(); break; } 
    case 2 : { example_2(); break; } 
    case 3 : { example_3(); break; } 
    case 4 : { example_4(); break; } 
    case 5 : { example_5(); break; } 
    case 6 : { example_6(); break; } 
    default : return Usage(argv[0]);
    }

    cout << "...... Say goodbye to the Elephants." << endl;
    return 0;
}
```