**CSCE 2004**

**Lab 14: Practice Final Exam Solutions**

**Question 1) C++ Logic**

Consider the following declarations:

int catHeight = 6; // the cat's height

int dogHeight = 7; // the dog's weight

string dogName = "Rover"; // the dog's name

string catName = "Sylvester"; // the cat's name

float catWeight = 15.0; // the cat's weight

float dogWeight = 20.0; // the dog's weight

bool dogRabies = true; // flag which indicates whether animal has rabies

bool catRabies = false; // flag which indicates whether animal has rabies

**Answer the following questions by selecting the logical expressions that are**

**equivalent to the given English descriptions:**

The cat has rabies but does not weigh 20 lbs or less.

* 1. (catRabies != true) && (catWeight >= 20)
  2. (catWeight < 20.0) || (catRabies == true)
  3. !(catRabies == true) && (catWeight <= 20)
  4. **!(catRabies == false) && (catWeight > 20.0)**
  5. None of the above

The dog does not weigh less than the cat but neither the cat nor the dog has rabies.

* 1. !(dogWeight < catWeight) && !((catRabies == true) &&(dogRabies == true))
  2. !(dogWeight >= catWeight) && ((catRabies == true) || (dogRabies == true))
  3. **(dogWeight >= catWeight) && ((catRabies == false) && (dogRabies == false))**
  4. (dogWeight != catWeight) && ((catRabies == true) && (dogRabies == true))
  5. None of the above

The cat height and the dog height are not 10.

* 1. !(catHeight == 10) || !(dogHeight ==10)
  2. (catHeight != 10) && (dogHeight >= 10)
  3. !((catHeight != 10) && (dogHeight != 10))
  4. **((catHeight != 10) && (dogHeight != 10))**
  5. None of the above

**Question 2) Loops**

Consider the following code:

int main()

{

int Sum = 0;

int Num = 1;

while (Sum < 10)

{

Sum = Sum + Num;

Num = Num + 1;

}

cout<<"Sum= "<<Sum<<endl<<"Num= "<<Num<<endl;

return 0;

}

How many times will the program execute the code inside the loop?

1. 0
2. **4**
3. 5
4. 10
5. None of the above

What will the output of this program be?

1. (a) Sum=0, Num=1
2. (b) Sum=6, Num=4
3. (c) Sum=10, Num=4
4. (d) Sum=15, Num=5
5. **(e) None of the above**

Consider the following C++ code:

int main()

{

int Control = 2;

int Sum = 0;

while (Control < 5)

{

for (int Inner = 1; Inner <= 2; Inner++)

{

Sum = Sum + Control;

}

Control = Control + Control;

cout<<"Control="<<Control<<endl<<"Sum="<<Sum<<endl;

}

cout<<"Control="<<Control<<endl<<"Sum="<<Sum<<endl;

return 0;

}

What will the output of this program be?

1. **Control=4, Sum=4**

Control=8, Sum=12

Control=8, Sum=12

1. Control=2, Sum=4

Control=4, Sum=12

Control=8, Sum=12

1. Control=2, Sum=2

Control=4, Sum=2

1. Control=4, Sum=4

Control=4, Sum=4

1. None of the above

Which of the following C++ statements would have same effect as the line ***"for (int Inner = 1; Inner <= 2; Inner++)"***?

* 1. int Inner = 1; while (Inner < 3)
  2. for (int Inner = 2; Inner > 0; Inner++)
  3. int Count = 2; while (Count-1 > 0)
  4. **for (int Count = 0; Count < 2; Count++)**
  5. None of the above

**Question 3) Function Tracing**

Consider the following C++ code:

double modifyValue(double Value)

{

Value = Value \* 2.0;

return Value;

}

double modifyResult(double Result)

{

Result = Result / 3.0;

if (Result < 10)

return Result;

else

return modifyValue(Result);

}

int main()

{

double Value, Result;

Value = 42;

Result = modifyValue(Value);

cout<<endl<<Value<<" "<<Result;// Position A

Result = 36;

Value = modifyResult(Result);

cout<<endl<<Value<<" "<<Result;// Position B

return 0;

}

What values will be printed at position A?

* 1. 42.0 42.0
  2. 42.0 24.0
  3. 84.0 84.0
  4. **42.0 84.0**
  5. None of the above

What values will be printed at position B?

* 1. **24.0 36.0**
  2. 42.0 36.0
  3. 12.0 36.0
  4. 36.0 84.0
  5. None of the above

**Question 4) Function Design**

Consider the following C++ code:

// The purpose of the following function is to count the number of

// negative values in an array. Remember that zero is not negative.

// (eg. if the array contains {0, 2, -1, 4, -3, -7} the count is three).

int Process(int Array[])

{

int Count = 0;

// Iterate //

{

// Compare //

Count++;

}

return Count;

}

What should be in the place of // Iterate // to process the array?

* 1. for (int i = 0; i < 10; i++)
  2. for (int i = 1; i <= strlen(Array); i++)
  3. for (int i = 0; i < strlen(Array); i++)
  4. for (int i = 0; Array[i] < strlen(Array); i++)
  5. **None of the above**

What should be in the place of // Compare // to correctly increment count?

* 1. if (i < 0)
  2. if (Array[i] > strlen(Array))
  3. while (Array[i] < 0)
  4. if (Array[i] >= 0)
  5. **if (Array[i] < 0)**

**Question 5) Arrays**

This question will deal with Array declaration and array usage

1. If a programmer wants to use a 2-Dimentional array, which one of the following can be used to declair a 10 Rows and 5 Columns int array:
   1. int[10][5] array;
   2. int array[5][10];
   3. **int array[10][5];**
   4. int[5][10] array;
   5. None of the above
2. In the textbox below, write a function that takes a 2-Dimentional int array as an input and reterns the max value of that array and its location within the array

void FindMax(int array[10][5], int &max, int &row, int &col)

{

int maximum = array[0][0];

for(int i=0 ; i<10, i++)

{

for(int j=0 ; j<5 ; j++)

{

if(array[i][j] >= maximum)

{

maximum = array[i][j];

row = i;

col = j;

}

}

}

max = maximum;

}

**Question 6) File I/O**

Consider the following C++ code:

Consider the following text file called "data.txt":

1 student1 3.5

2 student2 0.6

3 student3 4.0

4 student4 2.2

5 student5 2.3

6 student6 3.8

7 student7 3.5

8 student8 3.1

9 student9 1.5

10 student10 3.0

11 student11 2.0

12 student12 1.6

13 student13 3.1

14 student14 3.9

15 student15 2.6

16 student16 4.0

17 student17 3.3

18 student18 2.6

Fill the blanks in the following code that will read the file, return a student with a specific ID

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

string idToLookFor;

string id;

string name;

string gpa;

cout << "What is the ID you are looking for";

cin >> idToLookFor;

ifstream Din;

Din.open("Data.txt");

if (!Din)

cout >> "Error. Unable to open file.\n";

else

{

while((Din >> id) && (id != idToLookFor))

Din >> name >> gpa;

if (id == idToLookFor)

{

Din >> name >> gpa;

cout << "The student you are looking for is " << name << ". His GPA is " << gpa << endl;

}

else

cout << "The student was not found." << endl;

Din.close();

}

return 0;

}

**Question 7) OOP Tracing**

Consider the following C++ code:

class Account

{

private:

int balance;

public:

Account () {balance = 100;}

int getBalance();

int updateBalance(int Amount);

};

int Account::getBalance()

{

return balance;

}

int Account::updateBalance(int amount)

{

balance = amount + 1;

return amount;

}

int main()

{

Account mySavings;

cout<<endl<<"My balance is: "<<mySavings.getBalance(); //A

int Amount = mySavings.updateBalance(10);

cout<<endl<<"My updated Balance is: "<<mySavings.getBalance();//B

return 0;

}

What value will be printed at position A?

* 1. 10
  2. **100**
  3. 101
  4. 110
  5. None of the above

What value will be printed at position B?

* 1. 10
  2. 100
  3. 101
  4. 110
  5. **None of the above**

**Question 8)**

Given the following class definition:

const int MAXSTUDENTS = 100;

class Group

{

public:

Group ();

bool FillInfo (const char Filename[]);

int GetNumStudents () const;

void GetNumStudents (int &Num) const;

void Print() const;

bool ChangeGPA (const char Name[], const float GPA);

private:

void SetNumStudents( const int Num);

string Names[MAXSTUDENTS];

float GPAs[MAXSTUDENTS];

int UAIDs[MAXSTUDENTS];

int NumStudents;

};

Given the following definitions:

Group RowingClub;

int MyUAID;

int NumStudents;

Which of the following calls uses correct syntax?   
  
  
**Correct**/Incorrect : NumStudents = RowingClub.GetNumStudents();   
  
Correct/**Incorrect** : RowingClub.FillInfo ("Data.txt");   
  
Correct/**Incorrect** : Print(RowingClub);   
  
Correct/**Incorrect** : MyUAID = RowingClub.UAIDS[7];   
  
Correct/**Incorrect** : if (ChangeGPA ("Susan", 4.0)) cout << "Perfect\n";   
  
**Correct**/Incorrect : RowingClub.GetNumStudents (NumStudents);   
  
Correct/**Incorrect** : RowingClub.SetNumStudents (15);   
  
Correct/**Incorrect** : RowingClub.GetNumStudents (15);

**Question 9) OOP Design**

Suppose you are given the following definitions (appropriately placed in header/implementation files, with libraries/namespaces included as necessary):

class Apple

{

public:

Apple();

bool readData(ifstream &Input);

void print();

private:

float weight; //e.g., 1.62

string color; //e.g., "red"

int deliciousness; //a scale from 1 to 10!

};

...

const int MAX\_APPLES = 200;

class Basket

{

public:

Basket();

bool FillArray(const string filename);

void print();

private:

Apple apples[MAX\_APPLES];

int numApples;

};

...

Apple::Apple()

{

weight = -1.0;

color = "NULL";

deliciousness = -1;

}

bool Apple::readData(ifstream &Input)

{

bool success;

if(Input >> weight >> color >> deliciousness)

success = true;

else

success = false;

return success;

}

void Apple::print()

{

cout << weight << endl;

cout << color << endl;

cout << deliciousness << endl;

}

...

Basket::Basket()

{

numApples = 0;

}

...

Write code to implement Basket's readArray() method as defined above:

bool Basket::readArray(const string filename)

{

bool success;

ifstream Din;

Din.open(filename.c\_str());

if(!Din)

{

cerr << "File I/O Error!" << endl;

success = false;

}

else

{

while(apples[numApples].readData(Din))

{

numApples++;

}

Din.close();

success = true;

}

return success;

}

Write code to implement Basket's print() method as defined above:

void Basket::print()

{

cout << "Printing Basket contents..." << endl;

for(int i = 0; i < numApples; i++)

{

cout << "Apple " << i+1 << ":" << endl;

apples[i].print();

}

}

**Question 10) Pointers: Basic Usage**

Trace the following program and write the output at in the textbox below:

#include <iostream>

using namespace std;

int main()

{

int a = 6;

int b = 5;

int c = 10;

int \*aptr = &a;

int \*bptr = &b;

int \*cptr = &c;

cout << "The value of a is: " << a << endl;

cout << "The value of b is: " << \*bptr << endl;

\*aptr = \*aptr + \*cptr;

cout << "The value of a is: " << a << ". The value of c is: " << c << endl;

return 0;

}

The value of a is: 6

The value of b is: 5

The value of a is: 16. The value of c is: 10