CSS 1 – Final Exam

**What would be the output of the following program segment? NOTE: Be *very* specific about the output, showing it *exactly* as it would appear on the output screen.**

int alpha = 7;

int answer;

while (alpha < 10)

{

for (int c = 0; c < 4; c++)

{

if (alpha % 2 == 1)

cout << alpha;

else

cout << "alpha" << endl;

}

alpha++;

cout << endl;

}



**What would be the output of the following program segment?**

int var1, var2;

for(var1 = 0; var1 < 5; var1++)

{

for(var2 = var1 + 2; var2 > 1; var2--)

{

cout << "&";

}

cout << endl;

}



**What would be the output of the following program segment?**

void testFunc(int, int&, int&);

int main() {

int a = 3, b = 5, c = 6;

testFunc(a,b,c);

cout << a << " " << b << " " << c << endl;

return 0;

}

void testFunc(int a, int& b, int& c) {

a = b \* c;

b = c \* 6;

c = 3 \* a;

return;

}



**Which of these is a valid identifier for a function?**

3\_numbers

Three numbers

three\_numbers

three\_#s

**How many times will the following loop execute, assuming *val* is an integer initialized with a 5?**

for (int num = 2; num != 100; num += 2) {

counter++;

if (num > pow(val,2))

break;

}

**Which of the following statements shows correct syntax for opening a data file for reading?**

ifstream fileIn;

fileIn.open(“myfile.txt”);

ifstream fileIn;

open(“myfile.txt”, r);

ofstream fileIn;

fileIn << “myfile.txt”;

iostream fileIn;

fileIn.open(“myfile.txt”);

**Functions can:**

Be used as building blocks to create new programs.

Return a result to the caller function.

Be reused any number of times.

Do all of the above.

**Which of the following is *false* about the following function prototype?**

void function\_A( void);

It does not receive any arguments.

It could have been written: void function\_A( );

It does not return a value.

It could have been written: function\_A(void );

**Overloaded functions *must* have**

different parameter lists.

different return types.

the same number of parameters.

the same number of arguments.

**The data type *bool***

can take on any expression as a value.

can take on values true and false.

can take on values -1, 0 or 1.

can only be used in a selection statement.

**If a *do...while* structure is used,**

an infinite loop cannot take place.

counter-controlled repetition is not possible.

the body of the loop will execute at least once.

an off-by-one error cannot occur.

**The expression**

**if (num != 65)**

***cannot* be replaced by**

if (num > 65 || num < 65)

if ( !( num == 65 ) )

if (num - 65)

if ( ! (num - 65) )

**Of the following, which is *not* a logic error?**

not placing curly braces around the body of an if that contains 2 statements

using == to assign a value to a variable

failing to initialize counter and total variables before the body of a loop

using commas instead of the two required semicolons in a *for* loop heading

**An array is *not***

a consecutive group of memory locations.

subscripted by integers.

made up of different data types.

None of the above. (a dynamic entity)

**Write a loop that will print out only the values in an array that are in order. For example, if the array, *ara*, is declared and initialized as**

**int ara[] = {1, 3, 8, 5, 0, 6, 9};**

**output would show**

**1 3 8 9**

#include <iostream>

using namespace std;

void printArray(int arr[4], int size) {

for ( int i = 0; i < 4; i++ ) {

cout << arr[i] << ' ';

}

cout << endl;

}

int main()

{

int arr[4] = { 6, 2, 5};

int i, j, temp;

cout << "The array is: ";

printArray(arr, 4);

for(i=0; i<arr[4]-1; i++)

for(j=i+1; j<arr[4]; j++)

if(arr[i] > arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

for(i=0; i<arr[4]; i++)

cout<<" "<<arr[i];

cout<<endl;

return 0;

}

**Which of the following declares an array represented as this matrix:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

int matrix[3],[5];

int matrix[3][5];

int matrix[5][3];

int matrix[5,3];

**Which of the following is the correct way to determine if a file opened correctly?**

if (in\_file.NOTopen())

if (in\_file.failed())

if (in\_file.open.fail())

if (in\_file.fail())

**Write a block of code to sort out an array of 12 floating point values.**

bool user\_topping(true);

if (favorite(fin, user\_topping) )

cout << "We have your topping winner!" << endl;

else

cout << "No lunch for you!" << endl;

}

OR

#include <iostream>

using namespace std;

// returns true if topping x and y are equal, false otherwise

bool topping(int x, int y)

{

return (x == y);

}

int main()

{

using namespace std;

cout << "Enter topping one: ";

int x;

cin >> x;

cout << "Enter another topping: ";

int y;

cin >> y;

bool user\_topping = topping(x, y);

if (user\_topping)

cout <<"We have your topping winner!"<<endl;

else

cout <<"No lunch for you!"<<endl;

return 0;

}

**Write a loop that swaps the values in first half of an array with the values in the second half.**

suppose we want to swap values in arrays arrayONE[0] and arrayTWO[1]:

int temp=arrayONE[0];

arrayONE[0]= arrayTWO[1];

arrayTWO[1]=temp;

**Rewrite the if-else structure below as a switch structure.**

if (x == 2)

cout << "x = 2";

else if (x == 3 || x == 4)

cout << "x = 3 or x = 4";

else

cout << "Gandalf";

SWITCH\_\_\_\_\_\_\_\_\_\_\_\_\_\_

switch (x) {

case 1:

cout << "x = 2";

break;

case 2:

cout << "x = 3 or x = 4";

break;

default:

cout << "Gandalf";

}

OR\_\_\_\_\_

switch (x) {

case 1:

case 2:

case 3:

cout << "x is 2, 3 or 4";

break;

default;

cout << "Gandalf";

}

**Write a *for* loop that prints the numbers from 1-100 backward (including 1 & 100).**

int main()

{

int counter=100;

for (int i=1; i<=100; i++) {

counter--;

cout << counter << " ";

}

}

**What is the output of the following code segment?**

int array[4][4], index1, index2;

for(index1=0;index1<4;index1++)

for(index2=0;index2<4;index2++)

array[index1][index2]=index1 + index2;

for(index1=0;index1<4;index1++)

{

for(index2=0;index2<4;index2++)

cout << array[index1][index2] << " ";

cout << endl;

}



**Write a block of code to fill an array of integers with 40 random numbers in the range 10 - 30.**

int main()

{

// declare variables

const int size=40;

int array[size]={0};

int i;

srand(time(0)); // seed random number

for (int i=1; i<=size; i++) // maximum of size==40 elements allowed

{

((array[i]=rand() % (30-10)+10)); // range 10 and 30

cout << array[i] << " "<< endl;

}

}

**What preprocessor directive is needed to use the *setw(int)* function?**

#include <iomanip> // std::setw

**The following is part of a program to find a target value in an array.**

**Identify the errors in the program below. Correct the errors.**

**Complete the program by adding code to make the program work as described.**

//Searches a partially filled array of nonnegative integers for a target value.

#include <iostream>

using namespace std;

const int DECLARED\_SIZE = 20;

void fill\_array(int a[], int size, int& number\_used);

int search(const int a[], int number\_used, int target);

int main( )

{

int arr[DECLARED\_SIZE],

list\_size,

target,

result;

char ans;

//user input

fill\_array(arr, DECLARED\_SIZE, list\_size);

do {

cout << "Enter a number to search for: ";

cin >> target;

result = search(arr, list\_size, target);

if (result == -1)

cout << target << " is not on the list.\n";

else

cout << target << " is stored in array position " << result << endl

<< "(Remember: The first position is 0.)\n";

cout << "Search again?(y/n followed by Return): ";

cin >> ans;

}

while ((ans != 'n') && (ans != 'N'));

{

cout << "Program ended.\n";

return 0;

}

//end of main program

}

void fill\_array(int a[], int size, int& number\_used)

{

int next,

i=0;

cout << "Enter " << size << "nonnegative numbers.\n"

<< "End the list with a negative number.\n";

cin >> next;

while ((next >=0) && (i<size))

{

a[i]=next;

i++;

cin >> next;

number\_used=i;

}

}

int search(const int a[], int number\_used, int target)

{

const int T=1,

F=0;

int i=0,

located=false;

while ((!located) && (i<number\_used))

{

if (target==a[i])

located=true;

else

i++;

if (located) // bool

return i;

else

return -1;

}

}