**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Score:\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ECE 2036 Test 1**

Open book and notes, Phones and Tablets allowed as E book readers, but no Internet Access or PCs.

1. (*5%)* Fill in the typical size (per textbook and slides) in bytes of the following C/C++ variable types:

int \_\_*\_4*\_\_ bytes float \_\_*4*\_\_\_\_bytes long long \_\_*8*\_*\_\_* bytes

double array[100] \_\_*800\_\_*\_\_ bytes C style string with 10 characters \_*11*\_\_ bytes (not C++)

1. (*5%)* What does every C++ program need so that “*cin”* and “*cout”* can be used directly (i.e., without typing anything else or adding more statements)?

*#include <iostream>*

*using namespace std;*

1. (*5%*) Where is “*break*” most commonly found in C/C++ programs and what does it do in this situation?

*At the end of each case in a switch statement. Each case then jumps to the end of the switch to exit.*

1. (*5%)* The \_\_\_*Template*\_\_\_\_\_\_\_\_\_\_\_\_\_\_ feature in C++ can be used to automatically generate multiple versions of functions or classes that use different types (or classes).
2. (*5%*) In a C++ Standard Template Library (STL) *vector*, how could a user program check array bounds and what other C/C++ language keywords must be used to produce customized array out of bounds error messages in this case?

Use the .*at* member function. Use it inside of a *try*…*catch* block to catch the exception thrown for “out of bounds”.

1. (*10%)* Write a C/C++ *void* function called *absolute* with one float pointer argument that changes the value of the argument to it’s absolute value back in the calling program. Show an example call with the float variable *fvar*. Assume *fvar* is already declared as a float.

*void absolute(float \*a)*

*{*

*if (\*a<0) \*a = -\*a;*

*}*

*//call*

*absolute(&fvar);*

1. (*10%)* Write a C/C++ code segment using a *for* statement with an integer loop control variable “*i*” that would automatically sum all of the values in an array of integers called *myArray*. The size of *myArray* is not provided. For full credit, it must work for **any size array without changes or recompiling** on any machine or compiler. Save the sum in an integer variable called *total* which your code must declare and initialize. The loop control variable, *i*, must go out of scope at the end of the *for* statement, but *total* must stay in scope outside of the *for* statement. *myArray* is already declared and initialized elsewhere.

*int total=0;*

*for(int i=0; i<(sizeof(myArray)/sizeof(myArray[0])); ++i) {*

*total = total + myArray[i];*

*}*

1. (*21%)* Write the output produced by the C++ program below exactly as it will appear on the output device.

*2 4
6 8*

*10 12*

*14*

**#include "stdafx.h" //used only on Windoxs OS**

**#include <iostream>**

**using namespace std;**

**class testobj**

**{**

**public:**

 **testobj(int y);**

 **void q(testobj y);**

 **int getx()**

 **{**

 **return x;**

 **}**

 **void setx(int y)**

 **{**

 **x = y;**

 **}**

**private:**

 **int x;**

**};**

**testobj::testobj(int y)**

**: x(y)**

**{**

 **x++;**

**}**

**void testobj::q(testobj y)**

**{**

 **x = y.x + getx() - 1 ;**

**}**

**int main(int argc, \_TCHAR\* argv[])**

**{**

 **testobj A(1);**

 **testobj B(0);**

 **int i = 1;**

 **B.setx(3);**

 **while (i <= 7)**

 **{**

 **cout << A.getx() << " ";**

 **if (i%2==0) cout<< endl;**

 **A.q(B);**

 **i++;**

 **}**

**}**

1. (*18%)* Write the output that is produced by the C/C++ program below. Assume array “a” starts at address 0x002DF804 and that the compiler prints out all 32-bit addresses in hexadecimal.

*0x002DF804 249 9 0x002DF80C*

**#include "stdafx.h" //needed only in Windows OS**

**#include <iostream>**

**using namespace std;**

**int main(int argc, \_TCHAR\* argv[])**

**{**

 **int a[9]={0,1,2,3,4,5,6,7,8};**

 **int \*aptr;**

 **aptr = &a[0];**

 **a[2] = a[2] + 3;**

 **a[3] = a[2] + aptr[4];**

 **++aptr;**

 **(\*aptr)++;**

 **(\*(++aptr))--;**

 **cout <<&a<<" "<<a[1]<<a[2]<<\*(a+3)<<" "<<aptr[1]<<" "<<aptr;**

**}**

1. (*16% total*) The following two questions refer to the mbed C++ program attached at the end of this test. A pushbutton is connected from P23 to gnd, and a speaker with a driver transistor to P21.

(*2%*) What exactly is *funcD()* used for in this program?

*It displays the low four bits of the argument passed in the call on the mbed’s LEDs*

(*8%*) Describe exactly what the main program does from a user perspective. Include pushbutton operation, LEDs, events, sounds, and time in your answer. (a few sentences only).

*At power up it waits for the user to press the pushbutton. Every time the user presses the pushbutton, the following occurs;*

*It starts a for loop that runs 16 times that does the following:*

*Generates a new random number 0..15 using rand().*

*The random number plays a note on the speaker at a frequency of 200\*value for .1 seconds.*

*The random number is displayed in the LEDs in binary.*

*So the user sees fast flashing LEDs with different tones and it stops with a random number in the LEDs every time the pushbutton is pressed.*

(*4%*) What is the main purpose of the first *while* loop and explain how it achieves this?

*It waits for an initial pushbutton hit before starting and also sets up the random number generator. The amount of time it takes the user to hit the pushbutton to be hit the first time is measured with the timer. The timer value is used to seed the random number generator using srand(). This makes it start out with a different random number sequence every time it powers up.*

(*2%*) Why is the *wait(.01);* statement needed in this program?

*A small time delay is needed for the resistor pullup to take effect due to gate delays and capacitance. This avoids a phantom pushbutton hit when the device first powers up or after a reset.*

**#include "mbed.h"**

**#include "Speaker.h"**

**#include "PinDetect.h"**

**#include <math.h>**

**DigitalOut myLed1(LED1);**

**DigitalOut myLed2(LED2);**

**DigitalOut myLed3(LED3);**

**DigitalOut myLed4(LED4);**

**PinDetect pb1(p23);**

**Speaker mySpeaker(p21);**

**Timer t; //use a hardware timer**

**volatile int pbStatus = 0;**

**void pb1\_hit\_callback (void)**

**{**

 **pbStatus = 1;**

**}**

**void funcD( int i)**

**{**

 **myLed1 = i &0x01;**

 **myLed2 = (i>>1) & 0x01;**

 **myLed3 = (i>>2) & 0x01;**

 **myLed4 = (i>>3) & 0x01;**

**}**

**int main()**

**{**

 **unsigned int number=0;**

 **pb1.mode(PullUp);**

 **wait(.01);**

 **pb1.attach\_deasserted(&pb1\_hit\_callback);**

 **pb1.setSampleFrequency();**

 **t.start(); //start timer**

 **while(pbStatus==0) {};**

 **srand(t.read\_ms()); //read ms from timer**

 **while(1) {**

 **if(pbStatus==1) {**

 **pbStatus = 0;**

 **for(int i=0; i<16; ++i) {**

 **number = (rand() % 16);**

 **mySpeaker.PlayNote(200.0\*number,0.1,1.0);**

 **funcD(number);**

 **}**

 **}**

 **wait(.05);**

 **}**

**}**