

ECE 2036 Test 2 Practice Problems

1. ~~(5%)~~ A STACK is a common memory structure that enables recursion. It is used automatically by compilers to store subroutine or function return addresses along with any local variables.

2. ~~(5%)~~ The TEMPLATE feature in C++ can be used to automatically generate functions or classes that include different types.

3. ~~(5%)~~ What is the most common way in C/C++ to pass an entire 1D array as an argument to a function? For an example use "int array[10];", assume the array size can change, and use "void func1()" as an example function. Show the function prototype and an example function call.

void func1 (int [] , int)

FUNCTION PROTOTYPE ↑

func1(array, size);
FUNCTION CALL

4. If you automatically get an assignment operator with any class that you create, why would you ever want to overload the assignment operator for a given class?

You would overload if you had a pointer in the class to dynamic data.

5. C++ uses new and delete for dynamic memory management. If these do not match up during program execution so that any dynamically allocated memory is returned when no longer needed, a type of error occurs that is called a memory leak.

6. If a new class definition needs dynamic memory allocation, the user also typically needs to provide (i.e., not use the default) code for the classes'

Constructor,

copy constructor, and destructor. It is also likely that the user will want to overload the assignment operator (first before the other operators).

7. In a class that uses dynamic memory allocation for member data, assuming "A=B;" works some special code is also needed so that a statement like "A=A;" will work properly (i.e., A and B are objects from the new class).

This is called self assignment and code to handle it is typically

placed in the assignment operator overload.

Write a C++ statement in the space below that is typically added to solve the "A=A;" problem.

```
if (&RHS != this)
{
    // put in assignment code
}
return *this
```

- Will the following code produce a compile error? If so, please show a correction to the code so that it will compile correctly.

```
#include <iostream>
using namespace std;

class Happy
{

public:
    Happy (float level): HappyAmount(level), maxHappyAmount(10)
    {cout << "Welcome to Happiness" << endl;}

private:
    const int maxHappyAmount; // this constant needs to be
    float HappyAmount; // initialized in the initializer list
};

int main()
{Happy MyLevel(8);
return 0;}
```

placed in the assignment operator overload.

Write a C++ statement in the space below that is typically added to solve the "A=A;" problem.

```
if (&RHS != this)
{
    // put in assignment code
}
return *this
```

8. Will the following code produce a compile error? If so, please show a correction to the code so that it will compile correctly.

```
#include <iostream>
using namespace std;

class Happy
{

public:
    Happy (float level): HappyAmount(level), maxHappyAmount(10)
    {cout << "Welcome to Happiness" << endl;}

private:
    const int maxHappyAmount; // this constant needs to be
    float HappyAmount;       // initialized in the initializer list
};

int main()
{Happy MyLevel(8);
return 0;}
```

9. Write the output in the space below that is produced by the Constructor Destructor example C/C++ code provided with the test. Recall that most compilers use the copy constructor to make a new copy of the object whenever pass by value is used (instead of a pass by reference).

Assume this also includes functions that return a value.

A Def Con

B Int Con

B Copy Con

A +

A Int Con

-A Destructor

B Copy Con

B +

B Int Con

-B Destructor

-B Destructor

O 4

-B Destructor

-B Destructor

-A Destructor

```

// Constructors and Destructors Problem 9 Code
#include <iostream>
using namespace std;
class A{
public:
    A(); // Default constructor
    A(int); // int Constructor
    A(const A&); // Copy constructor
    ~A(); // Destructor
    A operator+(const A& rhs) const; // Addition operator
public:
    int x; // Single data member
};
A::A(): x(0){ cout<<"A Def Con      "<<endl; }
A::A(int x): x(x){ cout<<"A Int Con      "<<endl; }
A::~A(){ cout<<"-A Destructor      "<<endl; }
A::A(const A &a){
    x=a.x;
    cout<<"A Copy Con      "<<endl;
}
A A::operator+(const A& rhs) const
{
    cout<<"A +      "<<endl;
    A r(x + rhs.x);
    return r;
}
class B{
public:
    B(); // Default Constructor
    B(int); // int Constructor
    B(const B&); // Copy constructor
    ~B(); // Destructor
    B operator+(B rhs) const; // Addition operator
public:
    int x; // Single data member
};
B::B(): x(0){ cout<< "B Def Con      "<<endl; }
B::B(int x): x(x){ cout<< "B Int Con      "<<endl; }
B::~B(){ cout<< "-B Destructor      "<<endl; }
B::B(const B &b){
    x=b.x;
    cout<<"B Copy Con      "<<endl;
}
B B::operator+(B rhs) const
{
    cout<< "B +      "<<endl;
    return B(x + rhs.x);
}
int main()
{
    A a;
    B b(2);
    B c(b);
    a = a + a;
    b = b + c;
    cout << a.x << " " << b.x <<endl;
}

```

10. (25%) Write the output that is produced by this C/C++ program.

33953

```
#include <iostream>
using namespace std;

int main(int argc, _TCHAR* argv[])
{
    int a[8]={1,2,3,4,5,6,7,8};
    int *aptr;
    aptr = &a[0];
    a[2] = a[2] + 1;
    a[3] = a[3] + a[4];
    aptr++;
    (*aptr)++;
    (*(++aptr))--;
    cout << a[1] << a[2] << a[3] << a[4] << *aptr << endl;
}
```

11. (5%) Which of the following is NOT a valid storage class in c++

- a. register
- b. auto
- c. static
- d. kernal
- e. extern

12. (5%) Which of the following is a VALID declaration of an array of TEN double precision floating point numbers in C++?

- a. double array(10);
- b. double array(1,10);
- c. double array[10];
- d. double array[] = {1.0, 2.0, 2.71, 3.14, 4.0, 5.0, 6.0, 6.28, 9.9};
- e. none of the above are valid

13. (5%) A recursive function is one that calls itself.

14. (5%) Fill in the missing keywords and identifiers in the following *function template* that returns the maximum value of the three arguments of the same type.

```
template < typename T>
T maximum ( T value1, T value2, T value3)
{
    T maximumValue = value1;

    if (value2 > maximumValue)
        maximumValue = value2;

    if (value3 > maximumValue)
        maximumValue = value3;

    return maximumValue;
}
```

(at 6%) When passing the array of integers `listInts` to the method `printArray`, the copy of the entire array is a part of the argument list.

No, only the reference is passed to the function.

15. Consider the following code that has POSSIBLE syntax and logical errors. The intent of this program is to print of the entire contents of the array listNums.

```
#include <iostream>
using namespace std;

void printArray(int[]);

int main()
{
    // int listNums(10) = {1,2,3,4,5,6,7,8,9,10};
    int listNums[10] = {1,2,3,4,5,6,7,8,9,10};
    printArray(listNums);
    return 0;
}

void printArray(int nums[])
{
    for (int i=0; i<10; i++)
        // for (int i = 1; i <=10; i++)
        cout << nums[i] << endl;
}
```

syntax error

runtime error

- (a) (5%) Please correct any syntax errors that you see in this code.
- (b) (5%) Please correct any logical or runtime errors you see in this code.
- (c) (5%) When passing the array of integers listNums to the global function printArray, is a copy made of the entire array as a part of the function call?

No, only the pointer value is passed to the function.

16. (5%) In C/C++ we can have pointers. A pointer has as its value a(n) address.

17. (5%) Please identify the following operators assuming that they are in a C++ statement and NOT part of a declaration.

& address operator

* dereferencing operator

18. (5%) If you create a class called Game, do you need to overload the assignment operator (i.e. =) to make the assignment in the following snippet of code.

```
Game gameObject1;  
Game gameObject2;  
  
gameObject2 = gameObject1;
```

No! However if the object has a pointer then
you will need to overload the assignment
operator to get proper behavior.

19. (10%) In the following program, you need to add ONE character of text in multiple locations to make the output of this program be a 3. Please indicate the change in the program.

```
#include <iostream>
using namespace std;

void function1( int& );
int main( )
{
    int number = 5 ;
    function1 ( number );
    cout << number << endl;
    return 0;
}

void function1 ( int& num )
{
    num -= 2;
```

number
5 3

```
double quadratic( double x, double a, double b )  
{ return a*x*x + b*x; }
```

20. (10%) Please change the following code to have the function `double y(double x, double m, double b)` have default values for the slope (m) and y intercept (b) of 1 and 0, respectively.

```
#include <iostream>
using namespace std;

double y(double x, double m = 1.0, double b = 0.0);

int main()
{
    double x_value = 0;

    for (int i = 0; i < 20; i++)
    {
        cout << x_value << " " << y( x_value ) << endl;
        x_value = x_value + 0.1;
    }

    return 0;
}

double y(double x, double m, double b )
{
    return(m*x+b);
}
```

21. (20%) Show the output that is produced by this C++ program.

1.6 2
3.14159 2.71 2.71
2.71 3.14159 3.14159

```
#include <iostream>
using namespace std;

void crazyIvan(double [] );

int main()
{
    double dbArray[] = {1.6 , 2.71, 3.14159};
    double *ptr1 = NULL;
    double *ptr2 = NULL;

    ptr1 = dbArray;
    ptr2 = &(dbArray[1]);

    cout << *ptr1 << " " << (int) (*ptr2) << endl;

    ptr1 = ptr2 + 1;

    cout << *ptr1 << " " << *ptr2 << " " << dbArray[1] << endl;

    crazyIvan(dbArray);

    cout << *ptr1 << " " << *ptr2 << " " << dbArray[1] << endl;

} //end of main

void crazyIvan(double val[])
{
    double temp;

    temp = val[2];
    val[2] = val[1];
    val[1] = temp;
}
```

