CPS 271 Final Exam – Fall 2020 Name: _____Jinfang wei_____ [250 points]
Note: **TYPE** your answers, **attach** any code you write and the output, and return them.

1. Here is a definition of a class called small: [36 points]

```
class smallType
{
public:
    smallType();
    void k() const;
    void h(int i);
    friend void f(smallType z);
private:
    int size;
};
```

Suppose that x and y are both ***smallType*** objects. Write the word "YES" or "NO" in each location of this table to indicate whether the indicated statement is legal or illegal in these locations:

Statement	In a main program	In the const member function k	In the friend function f
`x = y;`	a. YES	b. NO	c.NO
`x.size = y.size;`	d. NO	e. NO	f NO
`x.size = 3;`	g. NO	h. NO	i. NO
`x.h(42);`	j. YES	k. NO	l. NO

2. Suppose that you define a new class called **foo**. For two **foo** objects x and y, you would like the expression x + y to be a new **foo** object. What is the prototype that you must write to enable expressions such as x + y? [3]

a) friend foo operator+ (const foo &);
b) friend foo operator+ (const foo &, const foo &);
c) friend void operator+ (const foo &);
d) foo operator+ (const foo &, const foo &) const;
e) None of the above.

3. I have written a class with an operator >> which is not a member function, but the operator >> implementation does access the private member variables of objects in the class. How did I define the operator to gain this access? [3]

a) friend ostream& operator>> (ostream&, const classname&);
b) friend istream& operator>> (istream&, const classname&);
c) istream& operator>> (istream&, classname&);
d) friend istream& operator>> (istream&, const classname&) const;
e) None of the above.

4. Is it possible for a member function of a class to activate another member function of the same class? [3]

a) No.
b) Yes, but only public member functions.
c) Yes, but only private member functions.
d) Yes, both public and private member functions can be activated within another member function.

e) None of the above.

5. Can two classes contain member functions with the same name? [3]

a) No.
b) Yes, but only if the two classes have the same name.
c) Yes, but only if the main program does not declare both kinds
d) Yes, this is always allowed, but only if they have different parameters.
e) None of the above.

6. Consider this class definition:

```
class quiz
{
   public:
     quiz();
     int f();
     int g() const;
   private:
     double score;
 };
```

Which function(s) can carry out an assignment `score = 1.0;`? [3]

a) Both f and g can carry out the assignment.
b) f can carry out the assignment, but not g.
c) g can carry out the assignment, but not f.
d) Neither f nor g can carry out the assignment.

7. The general syntax to overload the assignment operator = for a class is _____.[3]

a) `friend className& operator=(const className&);`
b) `className& operator=(className&);`
c) `string className& operator=(className&);`
d) `const className& operator=(const className&);`
e) None of the above

8. What is the primary purpose of a default constructor with default parameter? [3]

a) To allow multiple classes to be used in a single program.
b) To copy an actual argument to a function's parameter.
c) To initialize each object as it is declared.
d) To maintain a count of how many objects of a class have been created.
e) None of the above.

9. Suppose that the **foo** class does not have an overloaded assignment operator. What happens when an assignment a = b; is given for two foo objects? [3]

a) The automatic assignment operator is used
b) The copy constructor is used
c) Compiler error
d) Run-time error
e) None of the above

10. Which of the following blocks is designed to catch any type of exception? [3]

a) catch( ) { } b) catch (...) { } c) catch(exception) { } d) catch(*) { } e) none of the above

11. When should you use a const reference parameter? [3]

a) Whenever the data type might be many bytes.
b) Whenever the data type might be many bytes, the function changes the parameter within its body, and you do NOT want these changes to alter the actual argument.
c) Whenever the data type might be many bytes, the function changes the parameter within its body, and you DO want these changes to alter the actual argument.
d) Whenever the data type might be many bytes, and the function does not change the parameter within its body.
e) None of the above.

12. Consider the following declaration:

```
template <class T> class equality
{
    private:
     T x;
     T y;
};
```

Write a statement that shows the declaration in the class `equality` to overload the operator != as a member function? [3]

a) bool operator!= (equality s);
b) equality operator!= (equality s) const;
c) bool equality::operator!= (equality s1, equality s2) const;
d) bool equality::operator!= (const equality& s) const;
e) none of the above.

13. Here is a function prototype and some possible function calls:

```
int day_of_week(int year, int month = 1, int day = 1);
// Possible function calls:
cout << day_of_week( );
cout << day_of_week(1995);
cout << day_of_week(1995, 10);
cout << day_of_week(1995, 10, 4);
```

How many of the function calls are legal? [3]

a) None of them are legal
b) 1 of them is legal
c) 2 of them are legal
d) 3 of them are legal
e) All of them are legal

14. Given the following program: [3]

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

class Distance
{
```

```
    int feet, inches;
public:
    Distance(int f=0, int i=0):feet(f), inches(i)
    {}
    friend ostream & operator << (ostream & s, Distance &d);
};

ostream & operator << (ostream & s, Distance & d)
{
    s << d.feet << " - " << d.inches;
    return s;
}

void main()
{
    Distance d(1,2);
    ofstream out1("out1");
    ofstream out2("out2", ios::binary);
    out1 << d;
    out2.write(reinterpret_cast<char *>(&d),sizeof(Distance));
}
```

Which of the following is true about this program? circle the correct answer(s): [3]

a. Two files are left open because the close routines aren't called.
b. Because the buffers aren't flushed with a proper close call, the results are indeterminate.
c. The program properly creates 2 output files.
d. The program works only if the files out1, and out2 already exist. Otherwise opening non-existent files results in an error.
e. The reinterpret_cast call is an improper way to use casting.

Answer the next 4 questions (15 – 18) based on the following program:

```
// Assume normal headers exist
class MyException
{
public:
    string m_except;
    MyException (const char * str)    {
        m_except = str;
    }
};

class SpecialException : public MyException
{
public:
    int m_num;
    SpecialException(int num, const char * str):MyException(str)  {
        m_num = num;
    }
};
void throwIt(int t)
{
    try
    {
        switch (t)
        {
            case 0: throw 1; break;
            case 1: throw "hello"; break;
            case 2: throw MyException("hello"); break;
            case 3: throw SpecialException(1, "hello"); break;
            case 4: throw new MyException("hello"); break;
```

```
            }
        }
        catch (int i)  {
            cout << "throwIt int";
        }
        catch (SpecialException e)  {
            cout << "throwit Special";
        }
    }
    int main()
    {
        try    {
            //xxx
        }
        catch (int i)                  {  cout << "main int";    }
        catch (SpecialException e)     {  cout << "main SP";}
        catch (MyException e)          {  cout << "main Exception";   }
        catch (const char * str)       {  cout << "main const *";   }
        catch ( ... )                  {  cout << "main unknown";   }
    }
```

15. What is the output of the program if //xxx was replaced with throwIt(0)? [3]

a) main int b) throwIt int c) main Exception d) main unknown e) none of the above

16. What is the output of the program if //xxx was replaced with throwIt(1)? [3]

a) main const b) throwIt Speical c) main const * d) main SP e) none of the above

17. What is the output of the program if //xxx was replaced with throwIt(2)? [3]

a) main int b) throwIt int c) main Exception d) main unknown e) none of the above

18. What is the output of the program if //xxx was replaced with throwIt(3)? [3]

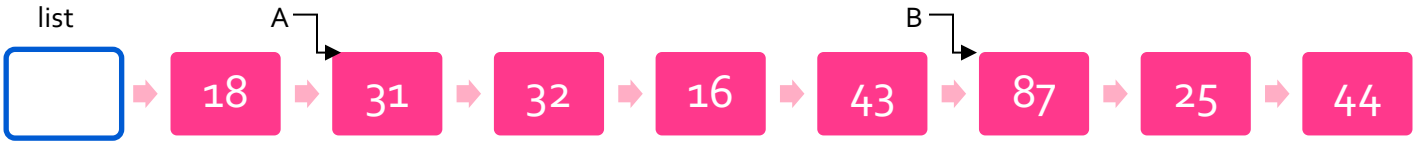
a) throwIt Special b) throwIt int c) main Exception d) main unknown e) none of the above

True or False questions (19 – 28): [30]

19. A pure virtual function is a virtual function that causes its class to be abstract. **True**
20. An abstract class is useful when no classes should be derived from it. **False**
21. A friend function can access a class's private data without being a member of the class. **True**
22. The keyword friend appears in the private or the public section of a class. **True**
23. A static function can be called using the class name and function name. **True**
24. A pointer to a base class can point to objects of a derived class. **True**
25. An assignment operator might be overloaded to ensure that all member data is copied exactly. **False**
26. A copy constructor is invoked when an argument is passed by value. **False**
27. The statements **virtual void sort(int);** and **void virtual sort(int);** are the same. **False**
28. Use **Car * parr[10];** to define an array called parr of 10 pointers to objects of class Car. **True**

29. How many parameters are required to overload the post-increment operator for a class as a friend function? [3]
    a) 0 b) 1 c) 2 d) 3 e) none of the above

Consider the following linked list. Assume that the nodes are in the usual info-link form. Use this list to answer the next 5 questions. (Assume all variables are defined) :



What is the output of each of the following: [15 points]

30. cout << list->info;

a) NULL b) 18 c) 32 d) 44 e) none of the above

31. cout << A->info;

a) 18 b) 31 c) 87 d) NULL e) none of the above

32. cout << B->link->info;

a) NULL b) 87 c) 25 d) 32 e) none of the above

33. cout << list->link->link->info;

a) 18 b) 32 c) 16 d) 87 e) none of the above

34.

```
p = list;
while (p != NULL)
{
        cout << p->info << "\t";
        p = p->link;
}
```

a) 18 b) 32 c) 16 d) 87 e) none of the above

35. Consider the definition of the following recursion function: [3 points]

```
int mystery(int first, int last)
{
      if (first > last)
            return 0;
      else if (first == last)
            return first;
      else
            return first + mystery(first + 1, last - 1);
}
```

what is the output of the following statement?

```
cout << mystery(6, 10) << endl;
```

a) 13
b) 21
c) 40
d) 42
e) none of the above

36. The class _____ is designed to deal with illegal arguments used in a function call: [3]

a. invalid_function

b. invalid_call
c. invalid_parameter
d. invalid_argument
e. none of the above.

37. Which of the following rules should you follow to solve the Tower of Hanoi problem? [3]
a. Only two disks can be moved at a time.
b. You can remove disks only from the first needle.
c. The removed disk must be placed on one of the needles.
d. A larger disk can be placed on a smaller disk.
e. none of the above.

38. Every node (except of the last node) in a singly linked list contains ____. [3]
a. the next node.
b. no address information.
c. the address of the next node.
d. the address of the previous node.
e. none of the above.

39. What is the purpose of the following code? [3]

```
current = head;

while (current != NULL)
{
    //Process current
    current = current->link;
}
```

a. Insertion of a node.
b. Selection of a node.
c. Traversal of a linked list.
d. Creation of a new list.
e. none of the above.

40. Which of the following is a basic operation on singly linked lists? [3]
a. Retrieve the data of an arbitrary node.
b. Swap the head and the last node.
c. Determine whether the list is full.
d. Make a copy of the linked list.
e. none of the above.

[41] List four common examples of exceptions: [4 points]

a. runtime_error: overflow_error and underflow_error
b. logic_error: invalid_argument, length_error and out_of_range.
c.bad_alloc
d.bad_cast

I will upload cpp, header, main files question from 42 to 51

[42] Consider the definition of the following function template: [6]

```
template <class   T>
class strange
{
        T  a;
        T  b;
};
```

a. Write a statement that shows the declaration in the class **strange** to overload the **operator ==** as a member function.

b. Write a statement that declares **sObj** to be an object of type **strange** such that the private member variables **a** and **b** are of type **int**.

c. Write the definition of the function **operator==** for the class **strange**, which is overloaded as a member function. Assume that two objects of type **strange** are equal if their corresponding member variables are equal.

[43] Write a function template palindrome that takes vector parameter and returns true or false according to whether the vector does or does not read the same forwards as backwards. (e.g., a vector containing 1, 2, 3, 2, 1 is a palindrome, but a vector containing 1, 2, 3, 4 is not). [5 points]

[44] Write a recursive function that takes as a parameter a non-negative integer and generates the following pattern of stars. For example, if the integer is 4, then the pattern generated is: [5 points]

```
      * *
    * * * *
  * * * * * *
* * * * * * * *
* * * * * * * *
  * * * * * *
    * * * *
      * *
```

[45]. What is the output of the following code fragment? [5]

```
int grades[] = {12, 34, 56, 34, 34, 78, 38, 43, 12, 25};
vector<int> vGr(grades, grades + 10);
vector<int> ::iterator location ;
int quiz[2] = {43, 56};
ostream_iterator<int> scr(cout, "\t");

copy(vGr.begin(), vGr.end(), scr);
cout << endl;
copy(quiz, quiz + 2, scr);
sort(vGr.begin(), vGr.end());
copy(vGr.begin(), vGr.end(), scr);
cout << endl;
location = find(vGr.begin(), vGr.end(), 34);
int x = location – vGr.begin();
cout << x << endl;
```

[46]. Assume a random access file was created with the following code:

```
ofstream fileout("hardware.txt", ios::out | ios::binary);
```

```
 for (char ltr = 'A'; ltr <= 'Z'; ltr++)
{
        char ch = static_cast<char>(rand() (26) + 65);
        fileout.write((char *)(&ch), sizeof(ch));
}

fileout.close();
```

Write a statement to reverse the first N, say 10, characters in the file and update that file? Assume the file is open. [5]
[47] Consider the following definition of the recursive function print. [5]

```
void print(int num)
{
    if (num > 0)
    {
        cout << num << " ";
        print(num - 1);
    }
}
```

What is the output of the following statement?

```
print(4);
```

[48] Consider the following definition of the recursive function: [5]

```
int puzzle(int start, int end)
{
    if (start > end)
        return start - end;
    else if (start == end)
        return start + end;
    else
        return end * puzzle(start + 1, end - 1);
}
```

What is the output of the following statement?

```
cout << puzzle(5, 10) << endl;
```

[49] Write the definition of the function template that swaps the contents of two variables of any type. [5]

[50] Overload the operator += for the class myString to perform the following string concatenation. Suppose that s1 is "Hello" and s2 is "there". Then, the statement:

```
s1 += s2;
```

should assign "Hello there" to s1, in which s1 and s2 are myString objects. [5]
//

[Part 2 – Programming Part – 44 points]

[51] Write a class called PhoneNumber. The class should have the following string data members, areaCode, exchange, and lineNum.
The class will have to overload the >> operator.

a) Input the entire phone number into an array. Test that the proper number of characters has been entered. There should be a total of 14 characters read for the phone number of the form (800) 555-1212.
b) The area code and the exchange do not begin with 0 or 1.
c) The middle digit of an area code is limited to 0 or 1.
d) The area code should be surrounded by parenthesis and there is a dash between the exchange and the line.
e) There should be only one space between the area code and the exchange

If none of the above operations results in an exception, copy the parts of the phone number into the PhoneNumber object's data members. If there is an exception, throw an exception, catch the exception, display each error and ask the user to enter another phone number.

Write a main program to test your class and use the following sample input file:

```
(877)123-4567
(808)  123-4567
(101) 555-0111
(809) 123-9990
809 123 9999
(606) 555-1212        nothing wrong with this number
(000) 000-0000
```

Make sure you upload .h, .cpp and a screen print of the output. No output, no grade.