

Object Oriented Programming

Final Exam

1	2	3	4	5	TOTAL

Student ID: _____ Student Name: _____ Signature: _____

Q.1) (30) Study the following real world electrical appliances using object-oriented programming approach (e.g. What are the relationships between them? Can you exploit the inheritance between any of them? What are the polymorphic methods? Are there any abstract classes? If so what are the abstract methods? etc.).

➤ Electrical Appliances

- Audio
 - Hi-fi
 - Radio
 - Walkman
- Refrigeration
 - Fridge
 - Freezer
- Telephone
 - Fixed
 - Cordless
 - Cellular

- All electrical appliances have a `turnOn` and `turnOff` method.
- All audio appliances have an `adjustVolume` method.
- All refrigeration appliances have an `adjustTemperature` method.
- All telephones that have a `dial` and `hangup` method.

Write the class declarations only.

Q.2) (6) **Complete** the following code so that the template function `getMax()` is called.

```
template<class T,int size>
T getMax(const T *arr){
    T mx=arr[0];
    for (int i=1;i<size;++i)
        if (mx<arr[i]) mx=arr[i];
    return mx;
}
int a[]={7,3,6,1,5};
```

`int myMax=` _____

Q.3) (15) What is the output of the following code?

```
template<class T>
T max(const T v1,const T v2){
    cout << "Template max is called" << endl ;
    if (v1>v2) return v1;
    return v2;
}

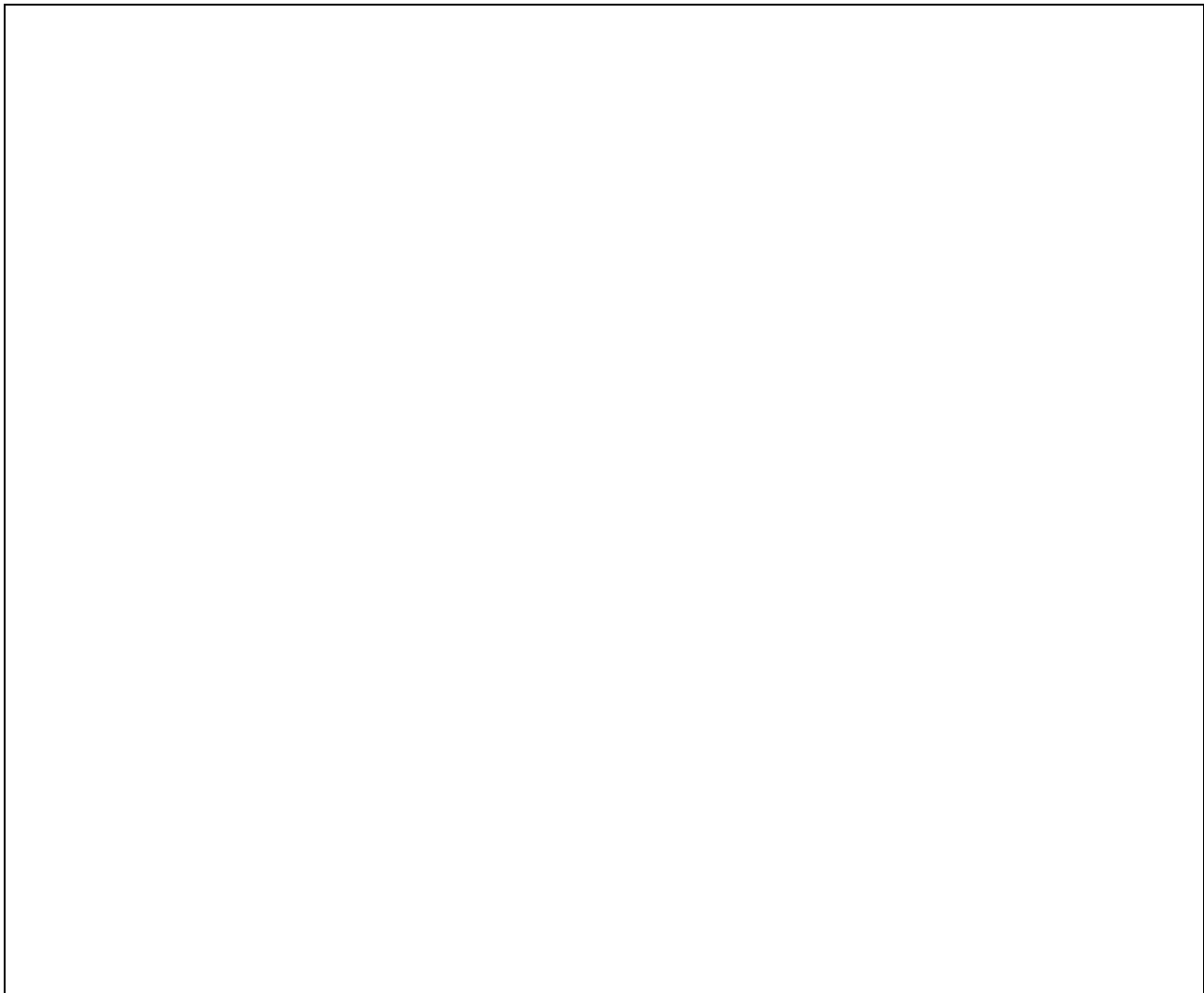
int max(const int v1,const int v2){
    cout << "Explicit max is called" << endl ;
    if (v1>v2) return v1;
    return v2;
}
max<int>(a,b);
max(a,static_cast<int>(b));
max(static_cast<long>(a),b);
max<double>(a,b);
max<double>(a,c);
```

1. _____
2. _____
3. _____
4. _____
5. _____

Q.4) (20) Design the necessary exception classes and **complete** the following code **considering** the main application code.

```
template <class T,int size>
class Stack {
    protected:
        T hold[size];
        int top;
    public:
        Stack(){top=0;}
        void push(const T& v) throw(_____);
        T& pop() throw (_____);
} ;
template <class T,int size>
void Stack<T,size>::push(const T& v) throw(_____) {
    if (_____) throw _____("Stack is full!");
    hold[top++]= v;
}
template <class T,int size>
T& Stack<T,size>::pop() throw(_____) {
    if (_____) throw _____("Stack is empty!");
    return hold[--top];
}
```

```
int main(){
    Stack<int,8> stack;
    int a[]={0,1,2,3,4,5,6,7,8,9};
    try {
        for (int i=0;i<sizeof(a)/sizeof(int);i++)
            stack.push(a[i]);
        for (int i=0;i<sizeof(a)/sizeof(int);i++)
            cout << stack.pop();
    }
    catch (StackOverflowException &e){
        cout << e.what() ;
    }
    catch (StackUnderflowException &e){
        cout << e.what() ;
    }
    catch (StackException &e){
        cout << e.what() ;
    }
    return 0;
}
```



Q.5) (32) What is the output of the following code?

```
class A {
public:
    int hold;
    A(int hold=0){this->hold=hold;}
    virtual void f(){
        hold+=7;
        cout << endl << "hold: " << hold ;
    }
    virtual void g()=0;
    void h(){
        hold+=11;
        cout << endl << "hold: " << hold ;
    }
};
```

```
class B {
public:
    int hold;
    B(int hold=0){ this->hold=hold; }
    virtual void f()=0;
    void g(){
        hold+=13;
        cout << endl << "hold: " << hold ;
    }
    void h(){
        hold+=17;
        cout << endl << "hold: " << hold ;
    }
};
```

```
int main(int argc,char *argv[]){
    D dd(2,4);
    A *ap = &dd ;
    B *bp = &dd ;
    D *dp = &dd ;
    cout <<      ap->hold;
    cout << endl << bp->hold;
    cout << endl << dp->hold;
    cout << endl << dd.hold;
    ap->f();
    ap->g();
    ap->h();
    bp->f();
    bp->g();
    bp->h();
    dp->f();
    dp->g();
    dp->h();
    dd.f();
    dd.g();
    dd.h();
    return 0;
}
```

```
class D : public A, public B {
public:
    int hold;
    D(int a=0,int b=0):A(b),B(b),hold(a) { }
    void f(){
        hold+=19;
        cout << endl << "hold: " << hold ;
    }
    void g(){
        hold+=23;
        cout << endl << "hold: " << hold ;
    }
    void h(){
        hold+=29;
        cout << endl << "hold: " << hold ;
    }
};
```

1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	
8		16	