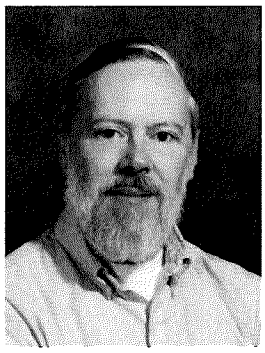
Printed Name SAHIL SINGH Net ID: SKS9379

# CS-UY 2124 - Object Oriented Programming
# MID-TERM EXAM #2 – November 21, 2023

- **THE BACK OF EACH PAGE IS TO BE USED AS SCRAP PAPER, IT WILL NOT BE SCANNED INTO GRADESCOPE NOR WILL IT BE GRADED!**
- **DO NOT SEPARATE ANY PAGE. (DO NOT PULL THIS TEST APART!)**
- **PRINT YOUR FULL NAME AS IT APPEARS IN ALBERT AT THE TOP OF EVERY PAGE.**
- This is a closed-book exam. No books, notes, calculators, computers, smart watches, or phones are allowed.
- Anyone found cheating on this exam will receive a zero for the exam
- If you have a question please ask the proctor of the exam.
- Note that we have omitted any `#includes` or `using namespace std;` statements in all questions in order to save space and to save your time thinking about them. You may assume that all such statements that are needed are present. And you don't have to write them either!!!
- You also do not need to write any comments in any of your code.
- Please read all questions carefully! They may look familiar and yet be completely different.
- Answering the short-answer questions, in particular, requires that you read and understand the programs shown. You need to read them carefully if you are going to understand them.
- If a question asks you to write a class or a function and provides you with test code, **be sure your class / function works with that test code.** If the question provides you with sample output, then your answer should match that output.
- Print your name and Net ID on the top of **EACH** page. (Yes, I know we already said that.)

Printed Name _____ Net ID: SKS97379

1. **EXTRA CREDIT (3 Points):** Who is the creator of the C programming language? Completely fill the circle next to your choice.



- O Larry Wall
- O Guido van Rossum
- O Ada Lovelace
- O John McCarthy
- O James Gosling
- O Bjarne Stroustrup
- ● Dennis Ritchie
- O Niklaus Wirth
- O Gary Kildall
- O None of the above

2. **(5 pts.)** Given the code below, what is the result of compiling and running the code? Completely fill the circle next to your choice.

```
const int SIZE = 8;
int main() {
      int* arr = new int[SIZE];
      for (int i = 0; i < SIZE; i++) {
          arr[i] = i*i;
      }
      int* p = arr + SIZE-4;
      int* q = p + 2;
      cout << "A: " << *p << ", ";
      cout << "B: " << q[-2] << endl;
}
```

0 1 4 9 16 25 36 49
0 1 2 3 4 5 6 7

- O The program compiles, runs and outputs: `A: 25, B: 9`
- O The program compiles, runs and outputs: `A: 16, B: 4`
- O The program compiles, runs and outputs: `A: 4, B: 16`
- ● The program compiles, runs and outputs: `A: 16, B: 16`
- O The program compiles, runs and outputs: `A: 9, B: 16`
- O The program compiles but has a run-time error
- O Compilation error
- O None of the above

Printed Name _____ Net ID: SK59379

3. **(5 pts.)** Given the code below, what is the result of compiling and running the code? Completely fill the circle next to your choice.

```
class Parent {
public:
    void display() const { cout << "Parent "; }          // Line A
};

class Child : public Parent {
public:
    virtual void display() const { cout << "Child "; }   // Line B
};

class Grandchild : public Child {
public:
    void display() { cout << "Grandchild "; }            // Line C
};

int main() {
    Child c;
    Grandchild gc;          // Line D
    Parent* par = &c;       // Line E
    par->display();         // Line F
    par = &gc;              // Line G
    par->display();         // Line H
}
```

Parent

- O The program will output: `Child Grandchild`
- O The program will output: `Parent Parent`
- O The program will output: `Child Child`
- ● The program will output: `Parent Child`
- O The program will compile and not output anything
- O The program will compile, but will crash when run
- O Compilation error at Line A
- O Compilation error at Line B
- O Compilation error at Line C
- O Compilation error at Line D
- O Compilation error at Line E
- O Compilation error at Line F
- O Compilation error at Line G
- O None of the above

Printed Name _____ Net ID: sk59779

4. **(5 pts.)** Given the code below, what is the result of compiling and running the code? Completely fill the circle next to your choice.

```
class Tile{
public:
    Tile(double len) :len(len) { }

    void computeArea() { display(len*len); }

    virtual void display(double area) const {
        cout << "Tile area: " << area << endl;
    }
private:
    double len;
};

class RoofTile : public Tile {
public:                                    2 2
    RoofTile(double len) : Tile(len) { }

    virtual void display(double area) const {
        cout << "Roof tile area: " << area << endl;
    }
};


int main() {
    Tile* tile = new RoofTile(2);
    tile->computeArea();
    delete tile;
}
```

O The program runs and outputs: `Tile area: 4`

● The program runs and outputs: `Roof tile area: 4`

O The program runs and outputs:
```
Tile area: 4
Roof tile area: 4
```

O The program runs and outputs:
```
Roof tile area: 4
Tile area: 4
```

O The program compiles but has a run-time error

O Compilation error

O None of the above

Printed Name _____ Net ID: SWS9979

5. **(5 pts.)** Given the code below, what will be output by compiling and executing the code? Completely fill the circle next to your choice.

```
class Parent {
public:
    virtual void display() const = 0;                    // Line A
};
void Parent::display() const { cout << "Parent"; }       // Line B


class Child : public Parent {
public:
    void display() const { cout << "Child "; }           // Line C
};

int main() {
    Child child;                                         // Line D
    Parent* pp = &child;                                 // Line E
    pp->display();                                       // Line F
}
```

- O Parent
- O Child
- O Compilation error at line A
- ● Compilation error at line B
- O Compilation error at line C
- O Compilation error at line D
- O Compilation error at line E
- O Compilation error at line F
- O None of the above

Printed Name _____ Net ID: SKS9979

6. **(5 pts.)** Given that the class `MyClass` defines the increment operators as members, what is the equivalent function call for the expression in the line marked "`THIS LINE`", below? Completely fill the circle next to your choice.

```
int main() {
    MyClass c;
    ++c;                        // THIS LINE
    c.display();
}
```

- O operator++(c)
- O c.operator++()
- ● c.operator++(0)
- O operator++(c,0)
- O all of the above
- O none of the above

7. **(5 pts.)** Given the following function definition:

```
void func(const string* const str_ptr) {
    // definition for the variable ptr goes here...

    ptr = str_ptr;
}
```

Which of the following definitions for the local variable `ptr` will allow the function `func` to successfully compile? Completely fill the circle next to your choice.

- O `string* ptr;` `// option 1`
- O `string const ptr;` `// option 2`
- O `string* const ptr;` `// option 3`
- O `const string& ptr;` `// option 4`
- ● `const string* ptr;` `// option 5`
- O `string ptr*;` `// option 6`
- O option 1 and option 3
- O option 3 and option 4
- O option 1 and option 5
- O None of the above

Printed Name _____ Net ID: SK9921

8. **(5 pts.)** Given the code below, what is the result of compiling and running the code? Completely fill the circle next to your choice.

```
class Pet {
public:
    Pet(string name) : name(name) {}                    // Line A
    virtual void speak() { cout << "Pet speaking ..."; } // Line B
private:
    string name;
};

class Cat : public Pet {
public:
    Cat(string name) : name(name) {}                    // Line C
    void speak() { cout << "Cat meowing ... "; }         // Linc D
};

int main() {
    Pet* ptr = new Cat("Felix");    // Line E
    ptr->speak();                   // Line F
}
```

- O The program compiles and runs, printing "`Pet speaking ...`"
- ● The program compiles and runs, printing "`Cat meowing ...`"
- O The program compiles and crashes when run.
- O The program compiles and runs without printing anything.
- O Compilation error at Line A
- O Compilation error at Line B
- O Compilation error at Line C
- O Compilation error at Line D
- O Compilation error at Line E
- O Compilation error at Line F

Printed Name _____ Net ID: _____

9. **(5 pts.)** Given the following code, what is the result of compiling and running the program? Completely fill the circle next to your choice.

```
class Dinosaur {
public:
    Dinosaur(string name) : name(name) {}
protected:
    string name;
};

class TRex : public Dinosaur {
public:
    TRex(string name) : Dinosaur(name) {}
    void attack(const TRex& trex) {
        cout << "TRex " << name << " attacking TRex " << trex.name;    // Line A
    }
};

class Raptor : public Dinosaur {    // Line C
public:
    Raptor(string name) :Dinosaur(name) {}

    void attack(const TRex& trex) {
        cout << "Raptor " << name << " attacking TRex " << trex.name;  // Line B
    }

    void attack(const Raptor& rap) {
        cout << "Raptor " << name << " attacking Raptor " << rap.name; // Line C
    }
};

int main() {
    TRex t("T");                                    // Line D
    Raptor rappy("Rappy");                          // Line E
    Raptor rippy("Rippy");                          // Line F
    rappy.attack(rippy);                            // Line G
}
```

- ● The program compiles, runs, and outputs: `Raptor Rappy attacking Raptor Rippy`
- ○ The program compiles, runs, and outputs: `Raptor Rippy attacking Raptor Rappy`
- ○ The program compiles, runs, and outputs: `Raptor Rappy attacking Trex T`
- ○ The program compiles and crashes when run.
- ○ Compilation error at Line A
- ○ Compilation error at Line B
- ○ Compilation error at Line C
- ○ Compilation error at Line D
- ○ Compilation error at Line E
- ○ Compilation error at Line F
- ○ Compilation error at Line G
- ○ None of the above

Printed Name _____ Net ID: JKS9379

10. **(5 pts.)** Given the following code, what is the result of compiling and running the program? Completely fill the circle next to your choice.

```
class Thing {
public:
    Thing(int val, int n = 0) : val(val), num(n) { }
    void display() { cout << val << ", " << num << endl; }
private:
    int val;
    int num;
};

int main() {
    Thing thingOne(6, 17);
    thingOne = 42;
    thingOne.display();
}
```

42, 0

- O The program runs and outputs: 6, 17
- O The program runs and outputs: 6, 42
- O The program runs and outputs: 17, 6
- O The program runs and outputs: 17, 42
- O The program runs and outputs: 42, 6
- ● The program runs and outputs: 42, 17
- O The program compiles but has a run-time error
- O Compilation error
- O None of the above

Printed Name Sahil Net ID: SKS977

11. **(5 pts.)** Given the following code, what is the result of compiling and running the program? Completely fill the circle next to your choice.

```
class Parent {
public:
    virtual void display() const { cout << "Parent "; }
};

class Child : public Parent {
public:
    void display() { cout << "Child "; }
};

class Gc : public Child {
public:
    void display() { cout << "Gc "; }
};

class GGc : public Gc {
public:
    void display() const { cout << "GGc "; }
};

int main() {
    vector<Parent*> vp;
    vp.push_back(new Parent);
    vp.push_back(new Child);
    vp.push_back(new Gc);
    vp.push_back(new GGc);
    for (Parent* ptr : vp) {
        ptr->display();
    }
}
```

Parent Child

- ● The program runs and outputs: `Parent Child Gc GGc`
- ○ The program runs and outputs: `Parent Parent Parent Parent`
- ○ The program runs and outputs: `Parent Parent Child Child`
- ○ The program runs and outputs: `Parent Parent Parent Child`
- ○ The program runs and outputs: `Parent Parent Parent Gc`
- ○ The program runs and outputs: `Parent Parent Parent GGc`
- ○ The program runs and outputs: `Parent Child Child Gc`
- ○ The program runs but immediately causes a run-time error
- ○ The program fails to compile

Printed Name Sahil Net ID: SKB9979

12. **(50 pts.)** The following problem involves 3 classes: **League**, **SL** and **Team**.
You are only responsible for implementing the **SL** (soccer league) class.

## League class

- has two fields: a string representing the name of the league, and a string representing its country.
- has a constructor accepting the name and country of the league
- supports copy control
- has getters to access those two fields, i.e. **get_name()** and **get_country()**
- **may have additional** fields and methods that you don't know about.
- **Do not assume** the existence of any other methods for the **League** class.
- **You are not responsible** for implementing the **League** class.

## Team class

- supports copy control and all necessary operators.
  - This includes relational operators (<, <=, ==, !=, > and >=) that compare two teams based on the number of points each has acquired.
- Teams have names, points, and a constructor to initialize them.
- **may have additional** fields and methods that you don't know about.
- **Do not assume** the existence of any other methods for the **Team** class.
- **You are not responsible** for implementing the **Team** class.

## SL class

- **SL** is a derived class of the **League** class.
- The **SL** class contains the following additional member variables:
  - ~~a `string` representing the country where the soccer league is played.~~
  - an `int` representing the division where the soccer league is played.
  - a `vector` of **Team** pointers
    - All of the **Team** objects are stored on the heap.
    - The **Team** objects are "owned by" the **SL** instance, i.e. no one else has a pointer to these **Team** objects.
    - The **Team** objects in the vector represent the league participants.
- A **constructor** that takes the **SL**'s name, country and division.
- **Copy control**. Yes, **all** of it! Copy assignment
- An **output operator**. You may choose the format, but the name, country, division and the teams in the soccer league should all be displayed clearly.
- a **`findLeagueLeader()`** method that returns a pointer to the team with the highest number of points. Returns null if there are no teams.
- An **`add_team`** method that takes in the team's name and points, creating the team object on the heap and adding it to the SL's vector. To save you a few lines of code, **you are not responsible for implementing this method. You will NOT use this method in your code.**

Implement the SL class so that it satisfies the above requirements
**and satisfies the test code on the following page.**

**[Test code and output are on the next page]**

Printed Name _____ Net ID: _____

## Test Code

```
int main() {
    SL slA("FreeLeague", "Austria", 17);
    SL slC("PoliteLeague", "Canada", 36);

    // Make sure all of this will work as shown!
    if (slA) {
      cout << slA.get_name() << " has teams!\n";
    } else {
      cout << slA.get_name() << " has no teams yet.\n";
    }

    slA.add_team("Moe", 2);
    slA.add_team("Larry", 5);
    slA.add_team("Curly", 3);

    cout << slA << endl;

    // Make sure all of this will work as shown!
    if (slA) {
      cout << slA.get_name() << " has teams!\n";
    } else {
      cout << slA.get_name() << " has no teams yet.\n";
    }

    cout << slA.get_name() << "'s leader is: "
       << *slA.findLeagueLeader() << endl;
}
```

## Sample output

```
FreeLeague has no teams yet.
FreeLeague, Austria, 17:
Team: Moe with 2 points
Team: Larry with 5 points
Team: Curly with 3 points
FreeLeague has teams!
FreeLeague's leader is: Team: Larry with 5 points
```

Begin your implementation of the **SL** class **on the next page**.

Printed Name _____ Net ID: _____

Begin your implementation of the **SL** class below.

```
#include <iostream>
#include <vector>
#include <string>

class League{ };


class SL:public League{
    public:
        friend ostream& operator<<(ostream& os, SL& rhs);
    SL(string name, string country, int division):
League(name,division),division(division){ }

    SL (const SL &rhs){
            League(rhs)
            name = rhs.get_name;
            country= rhs.get_county;
        division = rhs.division;
        for (auto &p : t){
            delete p;}
        t.clear()
        for (size_t i=0;i<(rhs.t).size();++i){
                SL* elem = new t[i];
                rhs.t[i] =&elem;}

    }
SL& operator= (const SL &rhs){
    if (*this!=&rhs){
            League::operator=(rhs);
                name = rhs.get_name;
                country = rhs.get_county;
                division = rhs.division;
        for (size_t i=0; (rhs.t).size(); ++i){
                SL*elem = new t[i];
                rhs.t[i] =&elem;}}}
```

Printed Name _____ Net ID: SKG9479

Continue your implementation of the **SL** class on this page.

```
~SL(){
    for(auto* elem : t){
        delete[] elem;
    }}
Void findLeagueLeader(){
    int high=0; string name;
    for(const auto& team m : t){
        if m.points > high{
            high = m.points;}
            name=m.name;}
        else{return nullptr;}
cout<< SL.getCountry<<"'s Leader is: Team"<<
    (*m).name<<" with "<< m.points
    <<" points"<<endl;
}
private:
    int division;
    int vector<Team*>t;
};
ostream& operator<<(ostream &os, SL&rhs){
    os<<rhs.get_name<<", "<<rhs.get_country<<"
    << rhs.division<<".";<<endl;
    if(&t){ for(auto& team* m: rhs.t{
        cout<<"Team: "<< m.name<<" with "
        << m.points<<" points"<<endl;
return os;
```

Printed Name _____ Net ID: _____

Continue your implementation of the **SL** class on this page.

Printed Name _____ Net ID: _____

Continue your implementation of the **SL** class on this page.