

# Midterm Exam 2

● Graded

Student

Total Points

89 / 100 pts

Question 1

Q1

3 / 0 pts

✓ + 3 pts H - Correct

Question 2

Q2

5 / 5 pts

✓ - 0 pts B - Correct

Question 3

Q3

0 / 5 pts

✓ - 5 pts D - Incorrect

Question 4

Q4

5 / 5 pts

✓ - 0 pts C - Correct

Question 5

Q5

0 / 5 pts

✓ - 5 pts B - Incorrect

Question 6

Q6

5 / 5 pts

✓ - 0 pts Correct

Question 7

Q7

5 / 5 pts

✓ - 0 pts H - Correct

Question 8

Q8

5 / 5 pts

✓ - 0 pts E - Correct

### Question 9

**Q9**

5 / 5 pts

✓ - 0 pts C - Correct

### Question 10

**Q10**

5 / 5 pts

✓ - 0 pts B - Correct

### Question 11

**Q11**

5 / 5 pts

✓ - 0 pts L - Correct

### Question 12

**Q12**

46 / 50 pts

Output operator

✓ - 0 pts Correct

Constructor

✓ - 0 pts Correct

Destructor

✓ - 0 pts Correct

Copy Constructor

✓ - 0 pts Correct

Assignment operator

✓ - 0 pts Correct

**computeTotalNumRiders**

✓ - 2 pts Data hiding: accessing RunData field directly

Operator bool

✓ - 2 pts Method should be **const**

Name \_\_\_\_\_ Net ID: \_\_\_\_\_

## CS-UY 2124 - Object Oriented Programming MID-TERM EXAM #2 - April 22nd, 2025

- **Do not open this test booklet until you are instructed to do so**
- Duration: 1 hour, 15 minutes
- Do not separate any pages. Do not pull the test apart from the staple.
- Ensure your **name and Net ID** is printed at the top of every page.
- This is a closed book exam, no calculators, computers, or phones are allowed
- Anyone found cheating on this exam will receive a zero for the exam
- Anyone who is found writing after time has been called will receive a zero for this 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, we said that already...).

Name \_\_\_\_\_

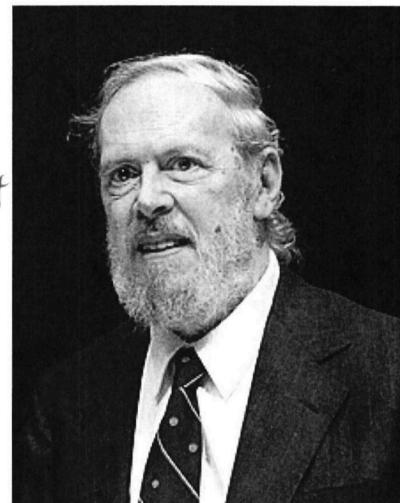
Net ID: \_\_\_\_\_

1. EXTRA CREDIT: Who is the creator of the C programming language? Write the letter that corresponds to your answer in the box provided.

- |                               |                                |
|-------------------------------|--------------------------------|
| A. Bill Gates                 | F. James Gosling <i>JG</i>     |
| B. Sergey Brin                | G. Bjarne Stroustrup <i>BS</i> |
| C. Guido van Rossum <i>Py</i> | H. Dennis Ritchie <i>DR</i>    |
| D. Ada Lovelace               | I. Claude Shannon              |
| E. Alan Turing                | J. None of the above           |

Your Answer

*H*



2. (5 points) Given the code below, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
class Entity {  
public:  
    void interact(int npcID) {  
        cout << "Entity interacts with NPC #" X override  
            << npcID << endl;  
    }  
};  
  
class Player : public Entity {  
public:  
    void interact(double lootValue) {  
        cout << "Player collects loot worth $" p 42.0  
            << lootValue << endl;  
    }  
};  
  
int main() {  
    Player p; p 49.99.  
    p.interact(42);  
    p.interact(99.99); e 7.  
    Entity* e = &p; (string)  
    e->interact(7);  
}
```

- |   |
|---|
| A. Entity interacts with NPC #42<br>Player collects loot worth<br>\$99.99<br>Entity interacts with NPC #7     |
| B. Player collects loot worth \$42<br>Player collects loot worth<br>\$99.99<br>Entity interacts with NPC #7   |
| C. Player collects loot worth \$42<br>Player collects loot worth<br>\$99.99<br>Player collects loot worth \$7 |
| D. Compilation error  |
| E. None of the above  |

Your Answer

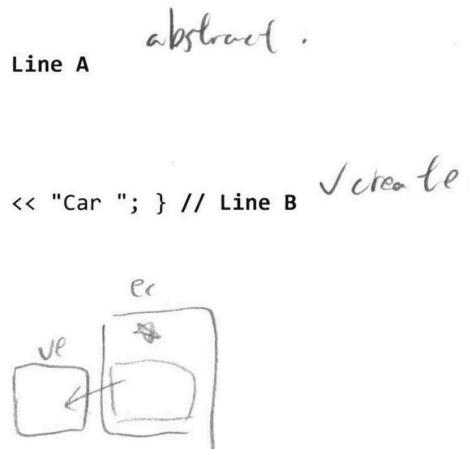
*B*

Name \_\_\_\_\_ Net ID: \_\_\_\_\_

3. (5 points) Given the code below, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
class Vehicle {  
public:  
    virtual void output_type() const = 0; // Line A  
};  
  
class Car : public Vehicle {  
public:  
    void output_type() const override { cout << "Car " } // Line B  
};  
  
class ElectricCar : public Car { };  
  
int main() {  
    ElectricCar ec;           // Line C  
    Vehicle* ecp = &ec;      // Line D  
    ecp->output_type();     // Line E  
    ec.output_type();         // Line F  
}
```

- A. Compilation error at line A
- B. Compilation error at line B
- C. Compilation error at line C
- D. Compilation error at line D
- E. Compilation error at line E
- F. Compilation error at line F
- G. It will compile, but will crash when run
- H. The program will output: **Car** **Car**
- I. The program will output: **Car** **ElectricCar**
- J. The program will output: **ElectricCar** **Car**
- K. The program will output: **ElectricCar** **ElectricCar**
- L. All of the above
- M. None of the above



Your Answer
D

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

4. (5 points) Given the code below, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
class Node {  
public:  
    Node() { cout << "1 "; }  
  
    ~Node() { cout << "2 "; }  
};  
  
class LinearStructure {  
public:  
    LinearStructure() { cout << "3 "; }  
  
    ~LinearStructure() { cout << "4 "; }  
  
private:  
    Node node;  
};  
  
class Stack : public LinearStructure {  
public:  
    Stack() { cout << "5 "; }  
  
    ~Stack() { cout << "6 "; }  
};  
  
int main() {  
    Stack my_data;  
}
```

1 3 5 6 4 2

- A. 1 3 5 2 4 6
- B. 1 3 5 4 2 6
- C. 1 3 5 6 4 2
- D. 1 5 3 2 4 6
- E. 1 5 3 4 2 6
- F. 1 5 3 6 4 2
- G. 3 1 5 2 4 6
- H. 3 1 5 4 2 6
- I. 3 1 5 6 4 2
- J. 3 5 1 2 4 6
- K. 3 5 1 4 6 2
- L. 3 5 1 6 4 2
- M. 5 3 1 2 4 6
- N. 5 3 1 4 2 6
- O. 5 3 1 6 4 2
- P. 5 1 3 2 4 6
- Q. 5 1 3 2 6 4
- R. 5 1 3 4 2 6
- S. 5 1 3 6 4 2
- T. None of the above

Your Answer
C

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

5. (5 points) Given the code below, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
void manipulate(const int& x) {  
    const int* p = &x; // Line A  
    int y = x; // Line B  
    cout << *p << ' '; // Line C  
    y = 53; // Line D  
}
```

```
int main() {  
    int y = 35;  
    manipulate(y);  
    cout << y << endl;  
}
```

- A. The program will have a compilation error at line A
- B. The program will have a compilation error at line B
- C. The program will have a compilation error at line C
- D. The program will have a compilation error at line D
- E. The program will have a runtime (or undefined) error at Line D
- F. The program will output 53 35
- G. The program will output 53 53
- H. The program will output 35 35
- I. The program will output 35 53
- J. All of the above
- K. None of the above

Your Answer

B

Name \_\_\_\_\_

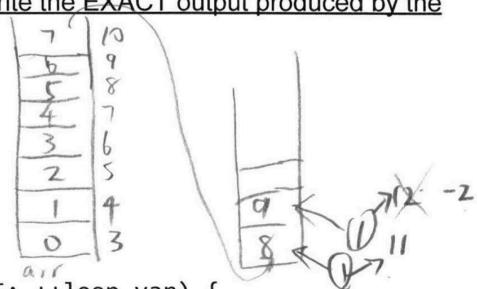
Net ID: \_\_\_\_\_

6. (5 points) What is the output of the program below. Write the EXACT output produced by the program in the box provided.

```
const int ARR_SIZE = 10;

int main() {
    int offset = -2;
    int* arr = new int[ARR_SIZE];
    for (int loop_var = 0; loop_var < ARR_SIZE; ++loop_var) {
        arr[loop_var] = loop_var + 3;
    }

    int* pointer_one = arr + ARR_SIZE - 1;
    int* pointer_two = pointer_one + offset;
    cout << "A: " << *pointer_two << endl;
    *pointer_one = offset;
    pointer_one--;
    cout << "B: " << *pointer_one << endl;
}
```



$$P_0 = arr + 9 \Rightarrow arr[9]$$
$$P_1 \neq arr[1]$$

Your Answer

A: 10  
B: 11

7. (5 points) Given:

```
int* quantities = new int[12];
```

which of the following is equivalent to <sup>quantities</sup>units[9]? There is only one correct answer!

- A. &(quantities\*9)
- B. &quantities+9
- C. &(quantities+9)
- D. (quantities+9)&
- E. quantities+9
- F. quantities+9&
- G. \*quantities+9
- H. \*(quantities+9)
- I. (quantities+9)\*
- J. quantities&+9
- K. quantities\*9
- L. None of the above

Your Answer

H

$$P[k] = * (P+k)$$

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

8. (5 points) Given the code below, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
class Surgeon;

class Doctor {
public:
    virtual void examine(Doctor& d) const { cout << "Doctor::examine(Doctor)"; }
    virtual void examine(Surgeon& s) const { cout << "Doctor::examine(Surgeon)"; }
};

class Surgeon : public Doctor {
public:
    void examine(Doctor& d) const override { cout << "Surgeon::examine(Doctor)"; }
    void examine(Surgeon& s) const override { cout << "Surgeon::examine(Surgeon)"; }
};

void performCheckup(Doctor& d1, Doctor& d2) { d1.examine(d2); }

int main() {
    Surgeon theSurgeon;
    Doctor theDoctor;
    performCheckup(theDoctor, theSurgeon);
}
```

- A. The program fails to compile
- B. A runtime error (or undefined behavior)
- C. The program runs and prints **Surgeon::examine(Doctor)**
- D. The program runs and prints **Surgeon::examine(Surgeon)**
- E. The program runs and prints **Doctor::examine(Doctor)**
- F. The program runs and prints **Doctor::examine(Surgeon)**
- G. None of the above

Your Answer

E

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

9. (5 points) Given the following code, which choice is true? Write the letter that corresponds to your answer in the box provided.

Call by  
Derived  
class  
only

```
class Astronaut {  
protected:  
    void collect_sample() {}  
};  
  
class MissionSpecialist : public Astronaut {  
public:  
    void complete_mission() {  
        collect_sample(); // line A  
    }  
};  
  
int main() {  
    MissionSpecialist ms;  
    ms.collect_sample(); // line B X  
}
```

- A. Both Line A and Line B will compile
- B. Line A will not compile but Line B will compile
- C. Line A will compile but Line B will not compile
- D. Both Line A and Line B will not compile
- E. None of the above
- F. All of the above

Your Answer

C

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

10. (5 points) Given the following code, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
class Printer {  
public:  
    void print() {  
        cout << "Printer: Default print" << endl;  
    }  
  
    virtual void status() {  
        cout << "Printer: Ready" << endl;  
    }  
};  
  
class LaserPrinter : public Printer {  
public:  
    void print() { X overload  
        cout << "LaserPrinter: HQ print" << endl;  
    }  
  
    void status() override {  
        cout << "LaserPrinter: Warming up" << endl;  
    }  
};  
  
int main() {  
    Printer* p = new LaserPrinter();  
    p->print();  
    p->status();  
}
```

P D  
L W

A. The program prints:  
LaserPrinter: HQ print  
LaserPrinter: Warming up

B. The program prints:  
Printer: Default print  
LaserPrinter: Warming up

C. The program prints:  
Printer: Default print  
Printer: Ready

D. The program prints:  
LaserPrinter: HQ print  
Printer: Ready

E. Compile-time error due to missing virtual  
on print()

F. None of the above

Your Answer

b

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

11. (5 points) Given the code below, what is the result of compiling and running the code? Write the letter that corresponds to your answer in the box provided.

```
class Thing {  
public:  
    Thing(int n = 0, string s = "Felix") { // Line A  
        str = s; // Line B  
        num = n;  
    }  
  
    void display() {  
        cout << str << ':' << num << endl; // Line C  
    }  
  
private:  
    string str;  
    int num;  
};  
  
int main() {  
    Thing thingOne(17, "Heathecliffe"); // Line D  
    thingOne = 28; Implicit made // Line E  
    thingOne.display(); // Line F  
}
```

(28, "Felix")

- A. Compilation error at Line A
- B. Compilation error at Line B
- C. Compilation error at Line C
- D. Compilation error at Line D
- E. Compilation error at Line E
- F. Compilation error at Line F
- G. The program runs and prints:  
Heathecliffe:0
- H. The program runs and prints:  
Heathecliffe:17
- I. The program runs and prints:  
Heathecliffe:28
- J. The program runs and prints:  
Felix:0
- K. The program runs and prints:  
Felix:17
- L. The program runs and prints:  
Felix:28
- M. The program compiles and runs  
but does not print anything
- N. The program compiles but  
crashes with no output
- O. None of the above

Your Answer

L

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

12. (50 points) Define a class **RollerCoaster**, which inherits from the class **ParkAttraction**.

#### **ParkAttraction class**

RC::PA

You are not implementing this class.

PA (const string& name, int dur){ getName() }

The **ParkAttraction** class has a constructor that takes a name (which is the name of an attraction (e.g. "Sky Mountain")) and the duration (the length of the ride in seconds).

**ParkAttraction** has one method, **getName()**. It also has any necessary operators and supports copy control.

You should not need to know anything more about the class **ParkAttraction**.

#### **RunData class**

class RDS  
const string& t s  
int total  
}

You are not implementing this class.

A **RunData** object holds

- the timestamp of the coaster's run (a string value that represents the date and time when the coaster commences a run ("launches")) and,
- the total number of riders on the ride when it runs.

The **RunData** class supports copy control, and all necessary operators.

## **RollerCoaster class**

**This is the class that you are responsible for!**

The RollerCoaster class derives from **ParkAttraction**. It contains the following:

- two fields
  - the **RollerCoaster**'s max speed in kilometers per hour (an integer), and
  - a vector of **RunData** pointers.
    - All of the **RunData** objects pointed to in the vector will be stored on the heap.
    - **RunData** objects are added with the **RollerCoaster** method **addRun**. You will see that we used it in the test code. **You will not be using it and you will not define it**. Just figure you needed to know it was there.
- a method: **computeTotalNumRiders** that returns the sum of the number of riders on the rides for all runs in the vector.
- **constructor** - the constructor should accept the name of the coaster, duration of the ride, and the max speed for the coaster.
- **copy control** - all of it! All copies should involve making deep copies of the data, don't just copy pointers.
- **output operator** - Implement an output operator. You define the format of the output, but obviously all of the information you have about your **RollerCoaster** should be displayed, including its name.
- **conversion operator** - See the test code example with **thorsHammer**

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

**Example Code:**

```
int main() {
    // Duration: 30, Max Speed 40 mph
    RollerCoaster cyc("Cyclone", 30, 40);

    cyc.addRun("Jul 12, 1960, 10:00", 20);
    cyc.addRun("Aug 19, 1970, 9:00", 25);

    cout << cyc << endl;

    cout << "Cyclone's total number of riders: "
        << cyc.computeTotalNumRiders() << endl;

    RollerCoaster cyc2(cyc);
    cyc2.addRun("Jun 28, 1980, 11:00", 30);
    doNothing(cyc2);
    cout << "cyc: " << cyc << endl
        << "cyc2: " << cyc2 << endl;

    cout << "cyc2's total number of riders: "
        << cyc2.computeTotalNumRiders() << endl;

    cyc = cyc2;
    cout << "cyc: " << cyc << endl;

    // Thors Hammer: 42 second duration, Max speed: 60mph
    RollerCoaster thorsHammer("Thors Hammer", 42, 60);
    cout << thorsHammer.getName() << " has ";

    if (thorsHammer) {
        cout << thorsHammer.computeTotalNumRiders() << " total riders" << endl;
    } else {
        cout << "no riders\n";
    }
} // main
```

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

**Example Output:**

Cyclone, max speed: 40

Run: time stamp: Jul 12, 1960, 10:00 with 20 riders  
Run: time stamp: Aug 19, 1970, 9:00 with 25 riders

Cyclone's total number of riders: 45

cyc: Cyclone, max speed: 40

Run: time stamp: Jul 12, 1960, 10:00 with 20 riders  
Run: time stamp: Aug 19, 1970, 9:00 with 25 riders

cyc2: Cyclone, max speed: 40

Run: time stamp: Jul 12, 1960, 10:00 with 20 riders  
Run: time stamp: Aug 19, 1970, 9:00 with 25 riders  
Run: time stamp: Jun 28, 1980, 11:00 with 30 riders

cyc2's total number of riders: 75

cyc: Cyclone, max speed: 40

Run: time stamp: Jul 12, 1960, 10:00 with 20 riders  
Run: time stamp: Aug 19, 1970, 9:00 with 25 riders  
Run: time stamp: Jun 28, 1980, 11:00 with 30 riders

Thors Hammer has no riders

Start your Q12 answer here and continue onto the pages that follow!

```
class RollerCoaster : public ParkAttraction {  
private:  
    int speed;  
    vector<RunData*> allRuners;  
    friend ostream & operator<<(ostream& os, const RollerCoaster& rhs);  
public:  
    RollerCoaster (const string& name, int dur, int speed):  
        speed(speed), ParkAttraction(name, dur) {}  
    int computeTotalNumRiders () const {  
        int result = 0;  
        for (const RunData* each : allRuners){  
            result += each->totalRider;  
        }  
        return result;  
    }  
    //big 3  
    RollerCoaster (const RollerCoaster& rhs):  
        speed(rhs.speed), ParkAttraction(rhs) {  
        for (const RunData* each : rhs.allRuners){  
            allRuners.push_back(new RunData(*each));  
        }  
    }  
} //copy constructor
```

Continue your Q12 answer here and the pages that follow!

```
~ RollerCoaster() {
    for (RunData* each : allRuners) {
        delete each;
    }
    allRuners.clear();
}

RollerCoaster& operator=(const RollerCoaster& rhs) {
    if (this != &rhs) {
        for (RunData* each : allRuners) {
            delete each;
        }
        allRuners.clear();
        ParkAttraction.operator=(rhs);
        speed = rhs.speed;
        for (const RunData* each : rhs.allRuners) {
            allRuners.push_back(new RunData(*each));
        }
    }
    return *this;
}
```

Continue your Q12 answer here and the pages that follow!

```
explicit operator bool() {
    return computeTotalNumRiders() != 0;
}

}; // class RC
```

```
ostream& operator << ( ostream& os, const RollerCoaster& rhs ) {
    os << rhs.getName() << ", max speed: " << rhs.speed;
    os << endl;
    for ( const RunData* each : rhs.allRuners ) {
        os << " " << *each << endl;
    }
    return os;
}
```

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

**Continue your Q12 answer here and the pages that follow!**

Name \_\_\_\_\_

Net ID: \_\_\_\_\_

**Continue your Q12 answer here and the pages that follow!**



A large, empty rectangular box with a thin black border, intended for students to continue their answer to Question 12 and any subsequent pages.