

CS-UY 2124 Exam 2



TOTAL POINTS

74.5 / 103

QUESTION 1

1 Question 1 3 / 3

✓ - 0 pts Correct

QUESTION 2

2 Question 2 5 / 5

✓ - 0 pts Correct (A:18 B:8)

QUESTION 3

3 Question 3 0 / 5

✓ - 5 pts Incorrect

QUESTION 4

4 Question 4 5 / 5

✓ - 0 pts Correct (choice #2: Child's status: okay..)

QUESTION 5

5 Question 5 5 / 5

✓ - 0 pts Correct (choice #7: A C E B)

QUESTION 6

6 Question 6 0 / 5

✓ - 5 pts Incorrect

QUESTION 7

7 Question 7 0 / 5

✓ - 5 pts Incorrect

QUESTION 8

8 Question 8 5 / 5

✓ - 0 pts Correct (choice #7: Compilation error at Line 3)

QUESTION 9

9 Question 9 0 / 5

✓ - 5 pts Incorrect

QUESTION 10

10 Question 10 0 / 5

✓ - 5 pts Incorrect

QUESTION 11

11 Question 11 5 / 5

✓ - 0 pts Correct (choice #7: Compilation error at Line C)

QUESTION 12

12 Question 12 46.5 / 50

Inheritance

✓ - 0 pts Correct

operator<<

✓ - 0 pts Correct

Constructor

✓ - 0 pts Correct

Destructor

✓ - 0.5 pts If ranged for and setting to nullptr, then should be reference,

Copy constructor

✓ - 1 pts Not initializing non-primitive field in
initialization list

Assignment operator

✓ - 2 pts Incorrect / missing return

operator[]

✓ - 0 pts Point*

operator bool

✓ - 0 pts Correct

find

✓ - 0 pts Correct

CS-UY 2124 - Object Oriented Programming

MID-TERM EXAM #2 - April 18th, 2023

- Duration: 1 hour, 15 minutes
- **THE BACK OF EACH PAGE IS TO BE USED AS SCRAP PAPER, IT WILL NOT BE SCANNED INTO GRADESCOPE NOR 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. 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
- 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.

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

Larry Wall

Guido van Rossum

Ada Lovelace

John McCarthy

James Gosling

Bjarne Stroustrup

Dennis Ritchie

Niklaus Wirth

Gary Kildall

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 = 10;
int main() {
    int* arr = new int[SIZE];
    for (int i = 0; i < SIZE; i++) {
        arr[i] = 2*i;
    }
    int* p = arr + SIZE-2;
    int* q = p - 4;
    cout << "A: " << p[1] << ", ";
    cout << "B: " << *q << endl;
}
```

The program compiles, runs and outputs: A: 16, B: 8

The program compiles, runs and outputs: A: 16, B: 12

The program compiles, runs and outputs: A: 18, B: 8

The program compiles, runs and outputs: A: 8, B: 6

The program compiles, runs and outputs: A: 9, B: 4

The program compiles but has a run-time error

Compilation error

None of the above

(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 SoftDrink {
public:
    virtual void display() const = 0; // Line A
};

class Soda : public SoftDrink {
public:
    void display() const override { cout << "Soda "; } // Line B
};

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

int main() {
    GingerAle ga; // Line D
    Soda* sd = &ga; // Line E
    sd->display(); // Line F
    ga.display(); // Line G
}
```

The program will output: GingerAle GingerAle

The program will output: Soda GingerAle

The program will output: Soda Soda

The program will output: GingerAle Soda

The program will compile and not output anything

The program will compile, but will crash 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

(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:  
    Parent() {}  
    virtual void test(const string& status) const {  
        cout << "Parent's status: " << status;  
    }  
};  
  
class Child : public Parent {  
public:  
    void test(const string& status) const override{  
        cout << "Child's status: " << status;  
    }  
};  
  
void check(const Parent& w, const string& status) {  
    w.test(status);  
}  
  
int main() {  
    Parent* ptr = new Child;  
    string stat_str = "okay..";  
    check(*ptr, stat_str);  
}
```

- The program runs and outputs: Parent's status: okay..
- The program runs and outputs: Child's status: okay..
- The program runs and outputs: Parent's status:
- The program runs and outputs: Child's status:
- The program compiles but has a run-time error
- Compilation error because there is no **Child** constructor
- Any other compilation error
- None of the above

. (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:  
    Parent(){ cout << "A " ; }  
    ~Parent(){ cout << "B " ; }  
};  
  
class Child : public Parent {  
public:  
    Child(){ cout << "C " ; }  
    ~Child(){ cout << "D " ; }  
};  
  
class GrandChild : public Child {  
public:  
    GrandChild(){ cout << "E " ; }  
    ~GrandChild() { cout << "F " ; }  
};  
  
int main() {  
    Parent* ptr = new GrandChild();  
    delete ptr;  
}
```

ACEFDB

ACEDB

ECAB

The program fails to compile.

The program compiles and outputs nothing when executed.

ECABDF

ACEB

ECADF

None of the above

. (5 pts.) Given the code below, what is the equivalent function call for the expression "c++" in the line marked "THIS LINE", below? Completely fill the circle next to your choice.

```
class MyClass {  
public:  
    MyClass& operator++(){  
        re++;  
        return *this;  
    }  
  
    MyClass operator++(int dummy){  
        MyClass old = *this;  
        re++;  
        return old;  
    }  
  
    void display(){  
        cout << "real: " << re << ", imaginary: " << im << endl;  
    }  
  
private:  
    int re=0, im=0;  
};  
  
int main() {  
    MyClass c;  
    c++;  
    // THIS LINE  
    c.display();  
}
```

operator++(c)

c.operator++()

c.operator++(0)

operator++(c,0)

all of the above

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.

const string* ptr; // option 1

option 1 and option 2

string const ptr; // option 2

option 3 and option 4

string* const ptr; // option 3

options 1 - 5

const string& ptr; // option 4

None of the above

string ptr*; // option 5

(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 Bridge {  
public:  
    virtual void close() { cout << "closing..."; } // Line 1  
};  
  
class SwingBridge : public Bridge {  
public:  
    void swing() { cout << "opening... "; } // Line 2  
};  
  
int main() {  
    SwingBridge* ptr = new Bridge(); // Line 3  
    ptr->swing(); // Line 4  
}
```

The program compiles and runs, printing "opening..."

Compilation error at Line 2

The program compiles and runs, printing "closing..."

Compilation error at Line 3

The program compiles and crashes when run.

Compilation error at Line 4

The program compiles and runs to completion without printing anything.

None of the above

Compilation error at Line 1

. (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:
    virtual void attack(Dinosaur& dino) {
        cout << "Dino attacking! "; // Line A
    }
};

class TRex : public Dinosaur {
    void attack(Dinosaur& dino) {
        cout << "TRex attacking! "; // Line B
    }
};

class Raptor: public Dinosaur { // Line C
};

void battle(Dinosaur& d1, Dinosaur& d2) {
    d1.attack(d2); // Line D Dino TRex att
    d2.attack(d1); // Line E
}

int main() {
    TRex t; // Line F
    Raptor rappy; // Line G

    battle(t, rappy); // Line H
}
```

The program compiles, runs, and outputs:
Dino attacking! TRex attacking!

Compilation error at Line B
Compilation error at Line C

The program compiles, runs, and outputs:
TRex attacking! Raptor attacking!

Compilation error at Line D
Compilation error at Line E

The program compiles, runs, and outputs:
TRex attacking! Dino attacking!

Compilation error at Line F
Compilation error at Line G

The program compiles and crashes when run.
Compilation error at Line A

Compilation error at Line H
None of the above

. (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 Room {
public:
    Room(double size) : size(200) {
        cout << "Room size: " << size << " ";
    }
private:
    double size;
};

class Suite {
public:
    Suite(double size) : size(400) {
        cout << "Suite size: " << size << " ";
    }
private:
    double size;
};

class Hotel {
public:
    Hotel() : suite(700), room(350){} // Line 1
private:
    Room room; // Line 2
    Suite suite;
};

int main() {
    Hotel hotel; // Line 3
}
```

- The program runs and outputs: Suite size: 750 Room size: 350
- The program runs and outputs: Suite size: 400 Room size: 200
- The program runs and outputs: Room size: 350 Suite size: 700
- The program runs and outputs: Room size: 200 Suite size: 400
- The program compiles but has a run-time error
- Any other compilation error
- None of the above

.. (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 Pet {  
public:  
    virtual void eat() = 0;           // Line A  
};  
  
void Pet::eat() { cout << "Pet::eat"; }  
  
class Cat : public Pet {  
public:  
    void eat() { cout << "Cat::eat"; } // Line B  
};  
  
int main() {  
    Pet* petP = new Pet();           // Line C  
    Cat* catP = new Cat();           // Line D  
    petP->eat();                  // Line E  
    catP->eat();                  // Line F  
}
```

The program will output: Pet::eat

The program will output: Cat::eat

The program will compile and not output anything

The program will compile, but will crash 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

All of the above

None of the above

.. (50 pts.) The following problem involves 3 classes: **Plot**, **ScatPlot** and **Point**. You are only responsible for implementing the **ScatPlot** class.

Plot class

Note: you are **not responsible** for implementing the **Plot** class. The **Plot** class

- has a private string representing the plot's title
- defines a constructor accepting the plot's title
- supports copy control
- has a `get_title()` method that returns the plot's title
- may have additional fields and methods

Again note: you are **not responsible** for implementing the **Plot** class

ScatPlot class

ScatPlot is a derived class of the **Plot** class used to represent scatter plot objects. The **ScatPlot** class contains the following member variables:

- a private string representing the color of points on the plot
- a private bool indicating whether or not the **Plot** is currently being displayed with a default value of false
- a private vector of **Point** pointers
 - All of the **Point** objects in the vector will be stored on the heap.
 - The **Point** objects are "owned by" the **ScatPlot** instance, i.e. no one else has a pointer to these **Point** objects.
 - The **Point** objects in the vector represent the points installed on the **ScatPlot**.
 - Do not be concerned with how they were inserted into the vector, they are just there.
 - The **Point** class supports copy control and all necessary operators.
 - Do not assume the existence of any other methods for the **Point** class.

Again note: you are **not responsible** for implementing the **Point** class.

Define the **ScatPlot** class. You are only responsible for implementing the **ScatPlot** class' functionality as follows:

Implement the Constructor

Implement a constructor which accepts the **ScatPlot**'s title, color of points, and display status.

Implement Copy Control

Implement all of the copy control.

Implement the Output Operator

Implement an output operator. You choose the format, but the title, color of points, and the points contained in the **ScatPlot** must all be neatly displayed.

[More on the next page]

Implement an Index Operator

Implement an index operator that can access a `Point` pointer in the `ScatPlot`'s points vector based on the value passed to an index parameter. The behavior of the index operator is as follows:

- In the case that the index parameter has a value greater than or equal to the number of `Points` in the vector, the function returns `nullptr`.
- Otherwise, the `Point` pointer at the index passed as the value of the parameter is returned.
- The vector of pointers cannot be modified using the index operator.

Consider the statement below:

```
const Point* pt = scat_plt[35];
```

If `scat_plot` contains 36 or more points, `pt` will be assigned the address of the `Point` at index 35 of the points vector. In the case that the points vector contains fewer than 36 points, `nullptr` will be assigned to `pt`.

Implement a `find()` method

The `find()` method accepts a `Point` object and returns true if the point is contained in the `ScatPlot` on which its invoked and false otherwise.

Implement the Boolean Operator

When evaluated in a boolean context, the `ScatPlot` object will indicate whether or not it is currently displayed. You do not have to implement any behavior that displays or hides the `ScatPlot` object.

The following code (`sp` is a `ScatPlot` object) will compile and result in the correct output:

```
if (sp) { cout << "Currently displayed"; }
```

Note that based on your implementation, the following statement will result in a compilation error.

```
cout << sp + 1;
```

Begin your implementation of the `ScatPlot` class on the next page.

Name _____

Net ID: _____

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

Name _____

Net ID: _____

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

Name _____

Net ID: _____

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