Alex Gaskins

Department of Electrical & Computer Engineering

EE 553 – Engineering Programming: C++

Fall 2022 – Quiz2

Note

Please justify all your responses as much as you can.

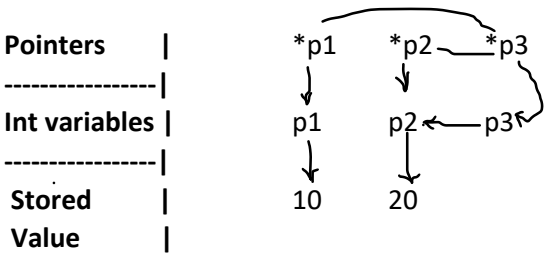
Total points 100

Question 1: Pointer (20 points)

Draw a boxes of the pointers and variables – use arrows to show the result of executing the following statements

```
int *p1, *p2,
*p3;  p1 = new
int;  *p1 = 10;  p2
= new int; *p2 =
20;  p3 = p2;
*p3 = *p3**p2**p1;
```

Determine the final values that pointers p1, p2, p3 are pointing to? Answer:



Question2: C++ Concepts - True/False (20 points)

1. The term "Overriding" means to redefine an existent method with different arguments. **F**
2. A class has a pure virtual method prototype is called abstract class **T**
3. In aggregation and association relationships, the outer class is responsible to create and destruct the inner class objects. **T**
4. The copy constructor is always called during the creation of an object **F**
5. The operator ?: can be overloaded using the operator overloading in c++ **F**

Question 3 C++ Classes and functions (20 points)

Define a class Vehicle with the following attributes.

Data attributes:

- Private: Seats = 40.
- Private: Wheels = 12.
- Private: Engine.

Operations attributes:

- Constructor to initialize the data attributes.
- Destructor to clear any data defined dynamically if any.
- Create virtual function called Show_specs() function to print all the vehicle information

Define a new class Sedan to inherit the class Vehicle in last question with the following modifications

- Private: Seats=4
- Private: Wheels=4 □ Private: Engine.
- Protected: color
- Protected: cc
- Public: brand

```
//Write Vehicle class code here

 #include <iostream>
using namespace std;

class Vehicle {

private:
  int seats;
  int wheels;
  int engine;
  // Number of cylinders (e.g. V6)

public:
  Vehicle() {
    this->seats = 0;
    this->wheels = 0;
    this->engine = 0;
  }

  Vehicle(int numSeats, int numWheels) {
    this->seats = numSeats;
    this->wheels = numWheels;
  }

  ~Vehicle() { cout << "\nCalling destructor" << endl; }

  virtual void setEngine(int numCylinders) { this->engine
= numCylinders; }
```

```
//Write Sedan class here

class Sedan : public Vehicle {

private:
  int seats;
  int wheels;
  int engine;

protected:
  string color;
  int cc;

public:
  string brand;

  Sedan() {
    this->seats = 0;
    this->wheels = 0;
    this->engine = 0;
    this->color = "";
    this->cc = 0;
    this->brand = "";
  }

  Sedan(int numSeats, int numWheels) {
    this->seats = numSeats;
    this->wheels = numWheels;
  }
```

```
  virtual void Show_specs() {
    cout << "\nVehicle info: \n";
    cout << "\n===============\n";
    cout << "Number of seats: " << this->seats << endl;
    cout << "Number of wheels: " << this->wheels << endl;
    cout << "Engine: " << this->engine << endl;
  }
};
```

```
  void setColor(string newColor) { this->color = newColor;
}
  void setCC(int theCC) { this->cc = theCC; }
  void setBrand(string theBrand) { this->brand =
theBrand; }

  void Show_specs() {
    cout << "\nSedan info: \n";
    cout << "\n===============\n";
    cout << "Number of seats: " << this->seats << endl;
    cout << "Number of wheels: " << this->wheels << endl;
    cout << "Engine: " << this->engine << endl;
    cout << "Color: " << this->color << endl;
    cout << "CC: " << this->cc << endl;
    cout << "Brand: " << this->brand << endl;
  }
};

//Write main function code here

int main() {

  Vehicle v;
  v.Show_specs();

  // 40 seats, 12 wheels
  Vehicle n(40, 12);
  n.Show_specs();

  // 4 seats, 4 wheels
  Sedan s(4, 4);
  s.Show_specs();

  s.setEngine(6);
  s.setColor("Blue");
  s.setCC(200);
  s.setBrand("Toyota");

  // Update sedan specs
  s.Show_specs();

}
```

Question 4 Pointers and Access Specifiers (20 points)

Given the below code, try to answer the following questions

```
#1  #include <string>
#2  #include <iostream>
#3  using namespace std;
#4  class Author
#5  {
#6  private:
#7  string name;
#8    int age;
#9
#10 public:
#11   Author() {}
#12   Author(string n, int y){name =n; age=y;}
#13   ~Author(){}
#14 void print() {
#15             cout << "Name: " << name << ", Age: "<< age << endl;
#16   }
#17 };
#18 class Book
#19 {
#20 private:
#21 Author *pA;
#22 public:
#23 Book() {}
#24 ~Book(){}
#25    void setAuth(Author *p) {
#26             pA = p;
#27     }
#28    void print() {
#29             cout << "("  << pA << ","<< endl;
#30    }
#31 };
#32 int main()
#32 {
#33   Author author1("Lily", 50);
#34 Book *pb = new Book;
#35 pb.setAuth(author1);
#36   delete pb;
#37 author1.print();
#38   pb->print();
#39
#40   if (1){
#41         Book b1;
#42         b1.setAuth(&author1);
#43    }
#44   author1.print();
#45 Book b2;
#46   b2.setAuth(&author1);
#47   author1.print();
#48 }
```

1. What is the relationship name between class Book and class Author?	Association
2. Line 35 has two compilation errors. Clarify the two bugs, and correct them.	- The variable **pb** is a pointer to book. Thus, it must either be specified as a pointer or use an arrow instead of a dot to point to its props. - **setAuth** takes one parameter that is a pointer to Author. In order to properly call this parameter, it must be able to be located by the pointer, so it must instead pass the address of the previously defined **author1** variable (&author1).
3. Line #43 has the end scope of object book1. What would happen to object author1 after line#43? Is it deleted from the memory?	**author1** will not be impacted, as it was defined outside of the conditional statement in which **b1** is defined.
4. Define a new object book3 of class book and assign a new author2 to book3	Author author2("Alex", 21); Book b3; b3.setAuth(&author2);

Question 5 C++ Concepts - multiple choices (20 points)

a. The dot operator connects which of the following two entities?

a) **Class object and member of that class** c) Class object and a class

b) Class and member of that class d) Class member and class object

b. A static function

a) should be called when an object is destroyed

b) **is closely connected with an individual object of a class**

c) can be called using the class name and function

d) Is used when a dummy object must be created

c. The break statement causes an exit:

a) **From the innermost loop or switch**

b) Only from the innermost switch

c) Only from the innermost loop

d) From all loops and switches

d. Which of the following can legitimately be passed to a function?
   a. A constant
   b. A variable
   c) A structure
   d) **All of these**

e. What is the Rvaule:
   a. The right-side of a binary
   b. Memory location associated with name
   c. The right most parameter in the function call
   d. **Memory location not associated with name**

f. What is the expected output of this Boolean expression: (1 == (-1 + 2)) == (41 && true)
   a. true
   b. **false**
   c) compilation error
   d) runtime error