THE UNIVERSITY OF WESTERN ONTARIO
LONDON CANADA

COMPUTER SCIENCE 3307A
FINAL EXAMINATION
DECEMBER 17, 2017
3 HOURS

NAME: _____

STUDENT NUMBER: _____

Question	
1-20.	_____
21-22.	_____
23.	_____
24.	_____
25.	_____
26.	_____
27.	_____
28.	_____
29.	_____
30.	_____
31.	_____
32.	_____
33.	_____
34.	_____
TOTAL	_____

(Out of 180 marks)

There are no cheat sheets, books, or other reference materials allowed for this exam. No calculators or other electronic devices are permitted either.

Part I -- Multiple Choice, True/False -- Choose the best answer from the choices given. Circle your answer on the paper. [40 marks total, 2 marks each]

1. The name C++ is derived from the ++ increment operator in C.
   a. True.
   b. False.

2. Before being called C++, the language was originally referred to as C with Classes.
   a. True.
   b. False.

3. The following is (are) true about C++.
   a. It is a dynamically typed language.
   b. It allows dynamic dispatch of functions.
   c. It allows inheritance.
   d. It performs garbage collection.
   e. More than one of the above statements are true.

4. Consider the following C++ code snippet:

```
1   int i = 4;
2   int j = 5;
3
4   int* k = &i;
5   int& l = j;
6
7   *k = 9;
8   l = 44;
9
10  cout << i << " ";
11  cout << j << " ";
12  cout << *k << " ";
13  cout << l << endl;
```

   What will be the output of the snippet? Assume `cout` and `endl` have been imported.

   a. `4 5 9 44`
   b. `4 5 4 5`
   c. `9 5 9 44`
   d. `9 44 9 44`
   e. The snippet will not compile.

5. The C++ language is a strict superset of C.
   a. True.
   b. False.

6. Classes are mandatory in all valid C++ programs.
   a. True.
   b. False.

7. A valid C++ program can have how many `main()` functions:
   a. 0.
   b. 1.
   c. More than 1.
   d. All of the above are, in fact, valid.

8. The following is the simplest statement in C++:

```
1  ;
```

   a. True.
   b. False.

9. In C++, a `class` and a `struct` are more-or-less equivalent, except for their default visibilities.
   a. True.
   b. False.

10. Which of the following is an acceptable way for a function to exit in C++:
    a. The function executes a `return` statement, providing an appropriate value as necessary, depending on the function.
    b. The function reaches the end of the function body, which is only allowed in `void` functions or `main()`.
    c. The function calls a system function that does not return (like `exit()`).
    d. All of the above.
    e. None of the above.

11. In C++, the destructor for an object is always called when the object is destroyed.
    a. True.
    b. False.

12. Friendship between classes in C++ is:
    a. Both reciprocal and transitive.
    b. Reciprocal but not transitive.
    c. Transitive but not reciprocal.
    d. Neither reciprocal nor transitive.

13. Static member functions in C++ have a `this` pointer.
    a. True.
    b. False.

14. Consider the following C++ code:

```
1  #include <iostream>
2  using namespace std;
3  class Parent {
4  public:
5     Parent() {
6        cout << "In Parent default constructor" << endl;
7     }
8     ~Parent() {
9        cout << "In Parent destructor" << endl;
10    }
11 };
12 class Child: public Parent {
13 public:
14    Child() {
15       cout << "In Child default constructor" << endl;
16    }
17    ~Child() {
18       cout << "In Child destructor" << endl;
19    }
20 };
21 int main() {
22    Parent p;
23    Child c;
24 }
```

What will be the output of the above code?

a.
```
In Parent default constructor
In Parent default constructor
In Child default constructor
In Child destructor
In Parent destructor
In Parent destructor
```

b.
```
In Parent default constructor
In Child default constructor
In Parent default constructor
In Parent destructor
In Child destructor
In Parent destructor
```

c.
```
In Parent default constructor
In Child default constructor
In Child destructor
In Parent destructor
```

d.
```
In Parent default constructor
In Parent destructor
In Child default constructor
In Child destructor
```

e. None of the above.

15. When overloading a function in C++, the definitions of the function must differ from each other by their:

    1. Name.
    2. Argument types.
    3. Number of arguments.
    4. Return type.

    Which of the following combinations of statements from the above list is correct:
    a. Statement 1.
    b. Statements 2 and 3.
    c. Statements 1, 2, and 3.
    d. Statement 4.
    e. None of the above options are correct combinations.

16. Which of the following statement(s) about user stories is/are correct:
    a. User stories are generated in a meeting with the client.
    b. They encourage deferring detail until later.
    c. They are intended to avoid "big design up front".
    d. They should be independent and estimatable.
    e. All of the above statements are correct.

17. When considering the use of aggregation or composition associations in UML, which of the following statements are true for the case of aggregation:
    a. An object comprised of many components owns those components.
    b. Components cannot exist independent of their owner.
    c. Components live or die with their owner.
    d. All of the above.
    e. None of the above.

18. The Prototype pattern, as discussed in class, works by wrapping one object inside another with the same interface.
    a. True.
    b. False.

19. The Bridge pattern, as discussed in class, defines a family of algorithms, encapsulates each one, and makes them interchangeable. Bridge lets the algorithm vary independently from clients that use it.
    a. True.
    b. False.

20. The Command pattern decouples the object that invokes an operation from the one that knows how to perform it.
    a. True.
    b. False.

Part II -- Fill In The Blank -- Indicate any compile-time or run-time problems with the following C++ code. Show output where appropriate. Write "ok" if there are no problems with the line in question. Note: simply writing "error" for a given line will not earn you any marks; you must indicate the nature of the error as well. [20 marks total]

21. Concerning `const` and primitive types [10 marks total, 1 mark per line]:

```
1  int i;
2  int j = 10;
3  const int l = 10;
4  const int m = 20;
5  const int n;
6  i = 5;
7  j = i;
8  m = l;
9  const int o = m;
10 int r = m;
```

Line 1 OK

Line 2 OK

Line 3 OK

Line 4 OK

Line 5 Error, must give a constant a value when declaring

Line 6 OK

Line 7 OK

Line 8 Can't change a constant once its already been assigned

Line 9 OK

Line 10 OK

22. Concerning pointers [10 marks total, 1 mark per line]:

```
1  int *p, *q, *r;
2  int c, d, e;
3  p = &c;
4  q = &d;
5  c = 10;
6  e = 0;
7  *r = e;
8  r = NULL;
9  *q = *p;
10 *p = *r;
```

Line 1 OK

Line 2 OK

Line 3 OK

Line 4 OK

Line 5 OK

Line 6 OK

Line 7 r isn't pointing to any memory, so it needs to be mallocs

Line 8 OK

Line 9 OK

Line 10 Segmentation fault

Part III -- Short Answer -- For the following questions, write your answer in the space provided, using diagrams as appropriate. You do not need to write full sentences, and can use point form if that is your preference. [60 marks total]

23. User stories help us to ensure that customer requirements are not misinterpreted by developers and that the customer gets what he/she actually wanted. Give two brief reasons why this is the case. [4 marks]

24. For each of the following user stories, indicate whether or not the story is acceptable according to the INVEST criteria we studied in class. If not, indicate which criteria the story violates. Briefly justify your answer. [6 marks total]
    a. *An agent should be able to manage tickets.* [3 marks]

    b. *The GUI must be aesthetically pleasing to the user.* [3 marks]

25. Provide a definition of a design pattern. Identify three benefits and three drawbacks of using design patterns in a software project. [8 marks]

26. For memory management, does C++ have built-in garbage collection? Provide one benefit and one drawback of C++'s approach to memory management. [6 marks]

C++ doesn't have a garbage collector, benefit is (good) developers have more control over memory, downside is (bad) developers have more control over memory

27. Complete the following table indicating the accessibility of various members of a class under different visibilities, using a "Yes" to indicate that the member is accessible in that situation or a "No" if it is not accessible. [9 marks]

	Public Member	Protected Member	Private Member
Accessible to members of its own class			
Accessible to members of a derived class			
Accessible to non-members (other code)			

28. Default parameters in C++ allow functions to be called without providing one or more trailing parameters, which can be convenient to software developers. Default parameters, however, are not without issues. Provide a scenario in which the use of default parameters leads to ambiguity when overloading a function in C++. [5 marks]

```
int aClass::doSomething(int i = 0)
{
   //DO SOMETHING
}
int aClass::doSomething()
{
   //DO SOMETHING
}
```

Causes problems b/c if there is no parameter given when the function doSomething, the compiler won't know which version of the function to use as when there is a default parameter and no parameter is passed when the function is called, it should pass in the default parameter, but since there is another version of the function that takes no parameters, the compiler won't know what to do.

29. In C++, what is the difference between a virtual method and a non-virtual method? Why is there a distinction between them in C++? [6 marks]

Virtual methods are from derived classes. The distinction is when a virtual method is created to override a method in the base class.

30. What is the definition of an abstract class in C++? Why are abstract classes useful? How can a class be made abstract in C++? [6 marks]

31. What is the smallest valid C++ program? Provide code for this program. [5 marks]

```
int main()
{
}
```

32. Write the smallest C++ program you know of that is guaranteed to cause a segmentation fault. (It is possible to do this by adding a single line of code to the program provided above.) [5 marks]

```
int main()
{
    printf("%s", 'p');
}
```

Part IV -- Long Answer -- For the following questions, write your answer in the space provided, using diagrams as appropriate. Again, you do not need to write full sentences, and can use point form if that is your preference. [60 marks total]

31. Recall the address book application created in the individual assignment earlier in this course. This application had a user facing interface that controlled the creation of an address book containing records of the names, addresses, and phone numbers for randomly generated people, as well as a REST interface for retrieving JSON-encoded address book information. [20 marks total]

a. Craft a user story capturing one of the requirements for this application. Use the front-of-card and back-of-card areas below to capture and record all of the requisite information for this user story. [8 marks]

Front of Card

Back of Card

b. Select one of the creational, behavioural, or structural design patterns discussed in class and discuss how you could apply this pattern to the application. In doing so, indicate how the use of this pattern would improve the design and quality of this application. [8 marks]

c. Provide a UML diagram, like those given in class accompanying our design pattern examples, to illustrate the use of and integration of your selected pattern from part b) above into the application. [4 marks]

32. In many design patterns, the power comes from having a set of classes with different purposes implement the same interface. For example, the Decorator pattern has `ConcreteComponent` and `Decorator` classes sharing the same `Component` interface. Identify two other design patterns requiring a common interface. For each pattern, describe the purpose (intent) of the pattern and give its UML diagram. Also, discuss what benefit the common interface provides in both patterns. [10 marks total]

33. Compare and contrast the Factory Method and Abstract Factory design patterns as discussed in class. Provide the appropriate class diagrams and address similarities and differences in intent and applicability. [10 marks]

34. In class, we introduced the SOLID design principles. Name and briefly describe each of these principles as well as their applicability to well-designed software systems. [10 marks]

This page has been left intentionally blank. Use it as additional workspace or extra space for answers if necessary.

This page has been left intentionally blank. Use it as additional workspace or extra space for answers if necessary.