CS 115 final prep

Object-Oriented Design (University of Regina)

CS 115 final prep

1. Is an object oriented programming language derived from C:
   - C++
2. Is a formal language that specifies a set/list of instructions for a compiler to execute?
   - Programming language
3. Has a list of binary instructions for a particular CPU:
   - Machine language
4. Has a language whose instructions are in the form of mnemonic codes and variable names:
   - Assembly language
5. Has a machine programming language that combines algebraic expressions and English words:
   - High level language
6. This indicates the start of a comment:
   - Double slash
7. This appears at the beginning of a program:
   - #include <filename>
8. This line of code indicates that our program uses objects defined in the namespace specified by region:
   - Using namespace std;
9. The body of the function is enclosed by what?
   - Curly braces
10. The function body should end with what?
    - Return 0;
11. These statements tell the compiler what data are needed in the function:
    - Declaration statements
12. These statements cause some action to take place when a program is executed:
    - Executable statements
13. These are used to name the data elements and objects manipulated by a program:
    - Identifiers
14. An identifier must always begin with what?
    - A letter or underscore symbol
15. An identifier can only consist of what?
    - Letters, digits, and underscores
16. You can't use what for identifiers?
    - Reserved words
17. Is a symbolic name for a memory cell that can be changed during the program?
    - Variable
18. Symbolic name for a value that can't be changed:
    - Constant
19. Is a set of value and operations that can be performed on those values?
    - Data types

20. What are the 4 data types in C++?
    - Integers (int)
    - Real numbers (float)
    - Booleans (bool)
    - Characters (char)
    - String
21. The bool data type has what two possible values?
    - True or false
22. Represents an individual character:
    - Char data type
23. Is a sequence of characters enclosed by quotation marks?
    - String
24. Ways data can be stored in memory?
    - Assigning to a variable
    - Reading data from an input device
25. An instruction that reads data from an input device into memory:
    - Input operation
26. Form of input operations:
    - Cin >> variable;
27. Form of output operations:
    - Cout << data element;
28. An instruction that displays info stored in memory:
    - Output operation
29. If all operands are of type int then the result is?
    - Int
30. If at least one operand is of type float then the result is?
    - Float
31. What are the two basic modes of computer operation?
    - Interactive and batch
32. In this mode, all must be supplied beforehand and the user cant interact with the program:
    - Batch mode
33. This mode lets the user interact with the program:
    - Interactive mode
34. Regulates the flow of execution:
    - Control structures
35. What are the three categories of control structures?
    - Sequence
    - Selection
    - Repetition
36. Is a group of statements bracketed by curly braces?
    - Compound statement
37. Means that each statement is executed in sequence:
    - Sequential flow

38. Is a control structure that chooses among alternative program statements?
    - Selection control
39. Repetition of steps in a program is called a?
    - Loop
40. This kind of expression has two possible values, true or false:
    - Logical expressions
41. What are the three logical operators?
    - && (and)
    - || (or)
    - ! (not)
42. If statement form:
    - If(condition)
    - Statements if true;
43. If-else statement form:
    - If (Boolean expression)
    - Statement if true;
    - Else
    - Statement if false;
44. What's a nested if statement?
    - If statements inside another with several alternatives
45. This statement is useful when the selection is based on the value of a single variable:
    - Switch control
46. Are modules that perform a task
    - Functions
47. Each function includes its own what?
    - Variables and statements
48. General form of a function:
    - Return_type Function_Name (parameters)
    - {
    - Declarations;
    - Statements;
    - Return expression;
    - }
49. Can functions return more than one value?
    - No
50. This process proceeds from the original problem at the top level to the subproblems at each lower level:
    - Top down design
51. Is a collection of data items stored under the same name:
    - Array
52. Are each individual element in an array:
    - Array element
53. How to declare an array?
    - Type name[size];

54. A value or expression enclosed in brackets after the array name
    - Array subscript
55. A variable followed by a subscript in brackets, designating a particular array element:
    - Subscripted variable
56. Two or more arrays with the same number of elements used to store related information about a collection of objects:
    - Parallel arrays
57. Multidimensional array declaration:
    - Type name[size1][size2]….;
58. How are arrays useful?
    - They can store a collection of data elements of the same type
59. How are structs useful?
    - They can store a collection of related items with different types
60. Do structs end with a semi colon?
    - Yes, because they are like prototypes or declarations
61. How can you access members of a struct?
    - Using a period placed between a struct variable and a member name for that struct type
62. How to pass a struct as a value to a function:
    - Void function(struct s)
63. How to pass a struct as a reference to a function?
    - Void function(struct & s)
64. They contain variables, that are operated on by instructions, selection, and loop statements:
    - object oriented programs
65. basic principles behind OOP:
    - objects
    - classes
    - inheritance
    - polymorphism
66. an object is what?
    - Any thing
67. A class consists of what?
    - A category of things
68. Is a specific item that belongs to a class?
    - an object
69. A class defines what?
    - The characteristics of its objects and functions that can be applied to its objects
70. What is the first part you create when creating a class?
    - Declaration section
    - Contains class name, variables, and function prototypes
71. What is the second part when you create a class?
    - Implementation section
    - Contains the actual functions
72. Making data fields private prevents what?
    - Outside manipulation of those fields

73. C++ classes are often split up into what 2 files?
    - Header files
    - Implementation of the class into .cpp files
74. The header file has what extension?
    - .h
75. Is a programming mechanism that binds together code and the date it manipulates?
    - Encapsulation
76. C++'s basic unit of encapsulation is what?
    - The class
77. Is a function that is called each time an object is created
    - Constructor
78. A constructor does what?
    - Initializes an object when its created
79. Do constructors have a return type?
    - No
80. When will you use a constructor?
    - When you need to give initial values to the instance variables defined by the class
81. Finding Min and Max Algorithm:

```
- Void FindMinMax(int array[], int size)
- {
-          Int min, max;
-          Min=max=array[0];
-          {
-                If(array[i] < min)
-                      Min=array[i];
-                Else if(array[i] > max)
-                      Max = array[i];
-          }
- Cout << …. << min << endl;
- Cout << …. << max << endl;
- }
```

82. Bubble swap algorithm:

```
-    Int main()
-   {
-       Int myarray[] = {…..};
-       Int tmp;
-        Bool swap;
-
-   Do
-   {
-    Swap = false;
-       For(int I = 0; I < size; i++)
-        {
-          If(myarray[i] >myarray[i+1])
-            {
-               Tmp = myarray[i+1];
-               Myarray[i+1] = myarray[i];
-              Myarray[i] = tmp;
-             Swap = true;
-            }
-        }
-   }while(swap);
-
-   For(int I = 0; I <= size; i++)
-   {
-     Cout << myarray[i] << “ “;
-   }
-     Return 0;
-   }
```

83. Searches a sorted array by repeatedly dividing the search interval in half:
    - Binary search
84. Is when a function calls itself:
    - Recursion
85. An object of class represents a single record in memory, if we want more than one record of class type, what do we do?
    - Create array of objects
86. Allows you to specify more than one definition for an operator or a function:
    - Operator & function overloading
87. The ability of an operator to perform different operations depending on the data type of its operands:
    - Operator overloading

88. Is a feature in c++ where 2 or more functions can have the same name but different parameters:
    - Function overloading
89. What's the form to overload an operator?
    - Return_type **operator** symbol (parameter list)
90. Are objects that represent sequence of characters:
    - Strings

91. Inheritance enhances what?
    - Reusability
92. The process of deriving new classes with additional data or new functions from existing classes
    - Inheritance
93. In order to derive a class from another we use what in the declaration of the derived class?
    - Colon
    - :
94. Form of derived class:
    - Class derived_class_name: public base_class_name
    - {
    - Statements;
    - };
95. A derived class can access what of its base class?
    - All the non-private members
96. Is a feature that allows a subclass or child class to provide a specific implementation of a function that is already provided by one of its super classes:
    - Overriding
97. When a child class function overrides a parent class function, you can say what?
    - You redefined the function
98. Is a member function you may redefine for other derived classes?
    - Virtual function
99. Tells the compiler that their function may be implemented later by inheriting class:
    - Virtual keyword
100. You can use this keyword to prevent classes from overriding
    - Final keyword
101. The constructor for each class in the derivation chain is called beginning with what?
    - The base class and ending with the most derived class
102. Is a function that is called automatically each time an object is destroyed:
    - Destructor
103. Destructors have the same name as what?
    - Their class

104.

How do you create a destructor class?

- Class Account
- {
- Public:
- ~Account();
- ….
- };

105.

The destructor for each class in the derivation is called beginning with what?

- The most derived class and ending with the base class

106.

Means having many forms:

- Polymorphism

107.

Polymorphism occurs when?

- There is a hierarchy of classes and they are related by inheritance

108.

C++ implements polymorphism through what?

- Overloaded functions
- Overloaded operators
- Virtual functions

109.

You can access the value of any variable by using what?

- The variables name

110.

Inserting what in front of the variables name allows you to access its address:

- An ampersand

111.

Declaring variables that can hold memory addresses are called what?

- Pointers

112.

A memory cell that stores the address of a variable or data object:

- Pointers

113.

You declare a pointer with a what?

- A type

114.

To indicate that a variable is a pointer, you use what following the data type?

- An asterisk
- *

115.

Often, we declare what to structs/ classes or to objects?

- Pointers

116.

When there is an unused memory space that cannot be allocated:

- Memory leak

117.

What operator is used to destroy dynamic variables?

- Delete

118.

The operator * is used to what?

- Declare a pointer and as well to access the memory space

119.

It's a special region of your computers memory that stores temporary variables by each function (including main)

- The stack

120.

Every time a function exits, all the variables pushed onto the stack are what?

- Freed (or deleted)

121.

Once a stack variable is freed, that region of memory becomes available for what?

- Other stack variables

122.

Is a region of your computers memory that is not managed automatically for you:

- The Heap memory

123.

To allocate memory on the heap, you must use what?

- The new operator

124.

Once you have allocated memory on the heap, youre responsible for what?

- Using **delete** to deallocate that memory once you don't need it anymore

125.

Is a storage pool of memory cells from which new storage is allocated whenever the **new** operator executes:

- The heap

126.

C++ memory map:

- STACK
- HEAP
- GLOBAL VARIABLES / PROGRAM CODE

127.

Is a simple yet powerful tool in C++

- Templates

128.

Syntax for a template function:

- Template <class T>
- T somefunction (T arg)
- {
- ….
- }

129.

Is a generic class for different types of objects:

- Class template

130.

Syntax for a class template:

- Template <class T>
- Class className
- {
- ….
- Public:
- T var;
- T someoperation(T arg);
- …..
- };

131.

How to create a class template object?

- className<datatype> classObject;
- usually put at the bottom of a file;

132.

Is a data structure in which only one element can be accessed:

- Stack

133.

Pushing onto a stack is called?

- Push

134.

A data structure in which the last element stored is the first one out:

- Last-In-First-out

135.

Popping a stack is called?

- Pop

136.

Compilers push a functions arguments onto a stack when what?

- A function is called

137.

Compilers also use stacks for data storage while what?

- Translating expressions

138.

In general, we use stacks in a program to remember what?

- A sequence of data objects or actions in the reverse order

139.

To use a stack, we need the compiler directive which is?

- #include <stack>

140.

Syntax for stack declaration:

- Stack <type> stack-name

141.

True or false: we declare a stack just like we declare an object of any template class:

- True

142.

A data structure that stores data on a last in, first out basis:

- A stack

143.

Describe the 2 basic operations on a stack:

- Push -> to add an element to a stack
- Pop -> to remove an element from a stack

144.

Is a data structure in which elements are inserted at one end and removed from the other end?

- Queue

145.

The queue is also referred to as what kind of data structure?

- First in first out structure (FIFO)

146.

To use a queue, you need what?

- The compiler directive
- #include <queue>

147.

How to declare a queue?

- Just like any template class
- Queue <type> queue_name;

148.

Member functions of the class queue?

- Push(const T)
- Top()
- Pop()
- Empty()
- Size()

149.
What does the push function do?
- Pushes its argument onto rear of queue

150.
What does the pop function do?
- Removes the element at the front of the queue

151.
What does the empty function do?
- Returns true if the queue is empty

152.
Is a collection of data items of the same type?
- Indexed list

153.
How does an indexed list grow?
- When you insert or append new elements

154.
The indexed list is what data type?
- An abstract data type

155.
The indexed list is an alternative to what?
- An array structure

156.
Is an indexed collection of elements just like an array
- The vector

157.
The vector class has many properties of what?
- The indexed list class

158.
How to use vectors?
- Need the compiler directive
- #include <vector>

159.
How to declare a vector?
- Vector<element_type> vector_name;

160.
If you omit size and the parentheses of a vector, whats happening?
- The initial size is set to 0

161.
Member functions of the vector class?
- Push_back
- Pop_back(int)
- Resize(int)

162.

One important concepts of OOP are what?

- Data hiding

163.

This mechanism is built in C++ that allows a programmer to access private or protected data from a non-member function and its done by using what?

- A friend function

164.

Is a function that is given the same access as methods to private and protected data

- Friend function

165.

Declaration of a friend function in c++

- Class className
- {
- …
- Friend return_type functionName (argument/s)
- …
- };

166.

True or false: you use the friend keyword in the definition/implementation:

- False

167.

Can two classes share a friend function?

- Yes

168.

This kind of class can access private and protected members of other classes in which it is declared in:

- A friend class

169. How to write a class which creates a stack class using the linked list:

```
struct node
{
 int data;
 node *link;
};

class MyStack
{
   node *top = NULL;
   node *nw;
   public:
      node* getTop()
     {
         return top;
      }
      void push(int d)
      {
         nw = new node;
         if(top==NULL)
         {
            nw->link = NULL;
         }
          else
          nw->link = top;
         nw->data = d;
         top = nw;
      }
      void pop()
       {
         if(top->link!=NULL)
        {
            node *to_del;
            to_del = top;
            top = to_del->link;
            delete to_del;
         }

         else
        {
            delete top;
         }
      }
```

```
    void display()
     {
       node *p = top;
       while(p != NULL)
       {
          cout << p->data << endl;
          p = p->link;
       }
    }
};
```

170. How to write a class which creates a queue class using the linked list:

```
struct node
{
 string data;
 node *link;
};

class MyQueue
{
   node *front=NULL, *nw, *back;
   public:
      node* getFront()
      {
         return front;
      }
      void push(string d)
       {
         nw = new node;
         if(front==NULL)
         {
            front = nw;
            nw->link = NULL;
         }
          else
            nw->link = NULL;
            back->link = nw;
         nw->data = d;
         back = nw;
      }
      void pop()
      {
         if(front->link!=NULL)
         {
            node *to_del;
            to_del = front;
            front = to_del->link;
            delete to_del;
         }
         else
         {
            delete front;
         }
      }
```

```
    void display()
     {
       node *p = front;
       while(p != NULL)
      {
          cout << p->data << endl;
          p = p->link;
       }
    }
};
```