Answer to The final of CS 135, Fall 2020

Name: Answer Student ID:

1. (1 point each, 10 points total) True or false.
   (1) To find out the square root of 4, we can use sqrt(4); **True**
   (2) An object is the blueprint from which classes are made. **False**
   (3) Encapsulation (also called information hiding) combines data members and method members and hide the implementation details from the clients of a class. **True**
   (4) In general, it is a good practice to access data members of one class from another class without using getter methods of the former class. **False**
   (5) Since each manager is an employee, we can design Manager class as a subclass of Employee class. **True**
   (6) A method can use a pointer to superclass as parameter. When we call this method, we cannot pass a pointer to subclass object of the superclass to that method. **False**
   (7) A static data member of a class is stored in every object. **False**
   (8) A virtual method declared in superclass can be overridden in subclassed. **True**
   (9) The size of one dimensional array arr can be found by arr.size(). **False**
   (10) In general, when we define a class, we should declare data member as public and operations on those data members as private. **False**

2. (2 points each) Choose the best solution for a multiple-choice problem.
   (1) Given string names[][3] = { {"Ann", "Bonny", "Cheryl"}, {"Abel", "Bob", "Charles"}}; What is names[1][0].size()? **Answer (B)**
   (A) 3
   (B) 4
   (C) 5
   (D) 6

   (2) What is vect.size() after the following code? **Answer (C)**

```
vector<int> vect;
vect.push_back(1);
vect.push_back(2);
vect.pop_back();
vect.push_back(3);
```

   (A) 0
   (B) 1
   (C) 2
   (D) 3

(3) Given int arr[] = {0, 1, 6}, what is the value of arr[3]? **Answer (D)**

(A) 0
(B) 1
(C) 6
(D) Invalid since the maximum index of arr is 2.

(4) Choose all valid statements. Which of the following is the most appropriate?

(A) int a = "hello";
(B) double b = 2;
(C) int c = (int) 2.6;
(D) Both (B) and (C)
(E) all of the above

Answer (D). Some students wonderful when we use int c = (int) 2.6. Suppose the total grade is double value 26.0 (some total grade can contain decimals, so the type is double). Then (26.0 / 10) is 2.6, but we only care about tens, so (int)2.6 converts 2.6 to 2. For example, total is double type, (int) (total / 10) returns an int and can be used in switch statement.

(5) What is the value of 1 / 2 * 3.0?

(A) 0
(B) 1
(C) 0.67
(D) 1.5

Answer (A). Note that / and * have the same precedence, and they are left aligned. Calculate 1 / 2 first. Note that this is integer division, it is like to divide one pen between two students, and we get 0, then 0 * 3.0 returns 0.
Aside: 1.0 / 2 * 3.0 returns 0.5 * 3.0 = 1.5 and 1 / (2 * 3.0) returns 1 / 6.0 = 0.167.

(6) Suppose there is a data member as int* called *grades*. What should be the code of void setGrades(int grades[], int size) looks like (choose all answers that may apply)? For simplicity, you do not need to check the validity of each element in grades. Do not need to worry about modification of data member size. **Answer (A)**

(A)
```
this->grades = new int[size];
    for (int i = 0; i < size; i++)
        this->grades[i] = grades[i];
```
(B) this->grades = grades;
(C) return grades;
(D) Either (A) or (B)

(7) Suppose Dog class extends from Animal class and has a default constructor. Which is correct?

Animal* a;
Dog* dog = new Dog();

(A) a = dog;
(B) dog = a;
(C) both (A) and (B)
(D) neither (A) nor (B)

Answer (A). Explanation: a pointer to animal type variable a can hold pointer to a dog, since each dog is an animal. It is like you can put a dog in a box marked by type "animal", but you cannot put an arbitrary animal object in a box marked by "Dog".

(8) Suppose we have Rectangle class, which has data member length and width. What can we say about the constructor of Rectangle?

(A) Constructor can have a name different from Rectangle.
(B) The return type of constructor can be void or other type.
(C) We can have constructor Rectangle(int length) and Rectangle(int width) at the same time.
(D) None of the above.

Answer (D). Option (C) is not correct. We cannot have two constructors with the same signatures (parameter list). Otherwise, when we call Rectangle rect(5); which constructor from Rectangle class should compiler choose?

(9) Road class has constructor to create a road with n blocks, where n is a positive int.

```
Road(int n)
```

How to declare and instantiate a Road object *road* with 80 blocks?

(A) Road road = 80;
(B) Road road = new Road(80);
(C) Road road(80);
(D) None of the above.

Answer (C). Note that (B) is not correct. It needs to be Road* road = new Road(80); Also (A) is not correct since 80 is an int. You cannot put an integer into a road object.

(10) Given API of getLastBlock method of Road class as follows. Suppose Road object rd has been properly declared and instantiated, how to find out the last block of rd?

```
int getLastBlock() const;
```

(A) int lastBlock = Road::getLastBlock();
(B) int lastBlock = rd.getLastBlock();
(C) rd.getLastBlock(int lastBlock);
(D) rd.getLastBlock(lastBlock);

Answer (B). Note that (A) is not correct since getLastBlock is not a static method from class Road.

(11) What is the best way to describe an array? **Answer (D)**

(A) The elements of an array can be put in non-adjacent memory location.
(B) The elements of an array can be of different types.
(C) We can only create an array of primitive type.
(D) None of the above.

(12) Suppose we write move method as follows,

```
void move() {
    int index = rand() % length;
    position += pattern[index];
}
```

Suppose the current value of data member position is 10, and data member pattern is an array with values [-1, 0, 1, 2]. Suppose when we call move method, rand() returns 10. What is the value of position after this calling?
Note that length is a data member to record the number of elements of pattern. It will be initialized in constructor.

(A) 9
(B) 10
(C) 11
(D) 12

Answer (C). Explanation: length of array pattern [-1, 0, 1, 2] is 4 and % is remainder operator, rand() % length returns 2. It is like to divide 10 pens among 4 students, each student gets the same number of pens, how many pens are left.

Note that patter is [-1, 0, 1, 2] and index is 2, so pattern[index] is 1. Current position is 10, increased by pattern[index] steps, the current position is 11.

(13) Suppose Dog class is inherited from Animal class. What is the proper head?

(A) class Dog
(B) class Dog : Animal
(C) class Animal : public Dog
(D) class Dog : public Animal

Answer (D)

(14) When we define a class inherited from some class, what should we do?

(A) Define constructors.
(B) Define setters and getters for data members unique to subclass.
(C) Override methods that are inherited from superclass but behaves differently from subclass to superclass.
(D) all of the above

Answer (D)

(15) What is the output of the following code? **Answer (D)**

```
int arr[] = {1, 2, 3, 4};
int* p = arr;
p += 3;
cout << *p << endl;
```

(A) 1
(B) 2
(C) 3
(D) 4

(16) Given the following code, **Answer (B)**

```
string message = "Hello";
char ch = message[1];
```

what is ch?

(A) Letter H
(B) Letter e
(C) Letter l
(D) Letter o

(17) How to release memory applied through in int* arr = new int[10]?

(A) arr = new int[10];
(B) delete[] arr;
(C) delete arr;
    arr = nullptr;
(D) delete[] arr;
    arr = nullptr;

Answer (D)

(18) What is the output of the following code? **Answer (D)**

```
int m = 5;
int n = 6;
m = n;
n = m;
```

(A) Variable m is 5, n is 6.
(B) Variable m is 6, n is 5.
(C) Variable m is 5, n is 5.
(D) Variable m is 6, n is 6.

(19) What is the output of the following code?

```
double grade = 68.9;
switch ((int)grade/10)
{
      case 10:
      case 9:
      case 8:
      case 7:
      case 6:
            cout << "pass" << endl;
      default:
            cout << "fail" << endl;
 }
```

(A) pass
(B) fail
(C) pass
    fail
(D) compilation error

Answer (C). We should have break; after the first cout statement.

(20) What is correct about array in C++?

(A) int[] arr = {1, 2, 3};
(B) int arr[] = {1, 2, 3};
(C) int arr[3] = {1, 2.6, 3};
(D) Both (B) and (C)

Answer (B). In (C), number 2.6 is of double type, not an int.

3. (10 points, 5 point each) Answer the following short answer questions.
   (1) Correct the errors, if any, in the code, which aims to swap two integers.

```
swap(int* m, int* n)
{
    int* temp = m;
    m = n;
    n = temp;
}

int main()
{
    int m = 5;
    int n = 6;
    swap(&m, &n);
    return 0;
}
```

Correction (either version is fine)

```
//Pass by reference
void swap(int& m, int& n)
{
    int temp = m;
    m = n;
    n = temp;
}

int main()
{
    int m = 5;
    int n = 6;
    swap(m, n);

    return 0;
}
```

```
//pass by pointer
void swap(int* m, int* n)
{
    int temp = *m;
    *m = *n;
    *n = temp;
}

int main()
{
    int m = 5;
    int n = 6;
    swap(&m, &n);

    return 0;
}
//warning: the following code
//just swap two pointers,
//it does not exchange the
```

	//values pointed by pointers.
	//int* temp = m;
	//m = n;
	//n = temp;

(2) What is the output of the following code? Suppose proper libraries are included, name spaces are properly used, and function prototype is declared.

```
int main() {
    for (int n = 1; n <= 10; n++)
        if (foo(n))
            cout << n << endl;
}

bool foo(int n) {
    if (n == 1)
        return false;

    //n does not equal 1
    //check whether [2, sqrt(n)] has a factor of n.
    //Once we find the first factor in [2, n/2], return false.
    for (int i = 2; i <= sqrt(n); i++)
        if (n % i == 0) //i is a factor of n that is < n
            return false;

    return true;
}
```

Function foo checks whether an int is prime or not. Method main print out all primes in [1, 10].

2
3
5
7

4. (5 points) **Define a method**, for a given array of strings, if every string in the array has at least 8 characters, then return true, otherwise (at least one string has fewer than 8 characters), return false.

```
#include <iostream>
using namespace std;

bool all_strings_size_lg_8(string arr[], int len)
{
    for (int i = 0; i < len; i++)
        if (arr[i].size() < 8)
           return false;

    return true;
}

//optional for this problem, for the purpose of competition
int main()
{
    string greetings[] = {"How are you?", "How do you do?",
        "Good morning.", "Hello"};
    int len = sizeof(greetings) / sizeof(greetings[0]);

    //display bool value in true / false using boolalpha
    cout << boolalpha
         << all_strings_size_lg_8(greetings, len) << endl;
    return 0;
}
```

5. (5 points) Define a two-dimensional array with 5 rows, the n-th row has n columns, where 1 <= n <= 5. The data of the array is as follows. Define and initialize the array, then print the array out as follows.

```
 0
 1  2
 3  4  5
 6  7  8  9
10 11 12 13 14
```

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    const int NUM_ROWS = 5;
    //Create a 2d array, n-th row has n columns,
    //where 1 <= n <= NUM_ROWS.
    int* arr[NUM_ROWS];
    for (int i = 0; i < NUM_ROWS; i++)
        arr[i] = new int[i+1];

    //Initialize elements of array.
    int k = 0;
    for (int i = 0; i < NUM_ROWS; i++)
        for (int j = 0; j < i+1; j++)
            arr[i][j] = k++;

    //Print 2d array in tabular format.
    for (int i = 0; i < NUM_ROWS; i++)
    {
        for (int j = 0; j < i+1; j++)
            cout << setw(2) << arr[i][j] << " ";
        cout << endl;
    }

    //Release memory.
    for (int i = 0; i < 5; i++)
    {
        delete[] arr[i];
        arr[i] = nullptr;
    }

    return 0;
}
```

6. (10 points) Define a class Triangle.

Each Triangle has three edges, let us call it a, b, c, where the size of any two edges is larger than the third one. For simplicity, assume the edges are integers.

(1) Declare class Triangle with data members and a constructor with three integer parameters.

(2) Define a constructor to take three parameters, if the given parameters are all positive and sum of any two parameter is larger than the third one, then use them to initialize the corresponding data members, otherwise, set a, b, c to 1.

```
class Triangle
{
public:
    Triangle(int a, int b, int c);

private:
    int a;
    int b;
    int c;
};

//Edges of triangle need to satisfy following requirements:
//(1) All edges are positive.
//(2) Sum of any two edges is larger than the third one.
//(3) Difference of any two edges is smaller than the third one.

//In fact, edges satisfy (2) also satisfy (1).
//For example, add two inequalities a + b > c and b + c > a,
//get a + 2b + c > c + a, so 2b > 0, or b > 0.
//Edges satisfy (2) also satisfy (3).
//From a + b > c, get a > c - b. From a + c > b, get a > b - c.
//In short, edges satisfy (2) form a triangle.
//For instance, 1, 2, 3 cannot form a triangle.
Triangle::Triangle(int a, int b, int c)
{
    if (a + b > c && b + c > a && a + c > b)
    {
       this->a = a;
       this->b = b;
       this->c = c;
    }
    else {
          this->a = 1;
          this->b = 1;
          this->c = 1;
    }
}
```

7. (10 points) Given Person class, extend it and define Doctor class.

Here are the signatures of the constructors and methods provided by Person class.

(1) Each person has a name (as a string) and age (as an int).
(2) Constructor Person() define name as "anonymous" and age as 18.
(3) Constructor Person(string name) uses given parameter name to initialized data member name, and initialize age as 18.
(4) Constructor Person(string name, int age) uses given parameter name to initialize data member age. If given parameter age is in the range of [0, 130], then uses given parameter age to initialize data member age, otherwise, set data member age to be 18.
(5) Method string getName() returns the name of a person.
(6) Method int getAge() returns the age of a person.
(7) Method void setAge(int age) uses given parameter age to reset data member age if the former is in the range of [0, 130], otherwise, leave the corresponding data member as it is.
(8) Method void setName(string name) uses given parameter name to reset data member name.

Each doctor is a person, just with additional data member to describe the name of all insurances he/she takes. This data member **insurances** can be a vector of string.

Your job: Declare Doctor class as a subclass of Person and define the following methods. NO NEED TO define Person class.

(1) Default constructor Doctor() with **insurances** initialized by "medicaid" and "medicare".
(2) Write a method, given a string representing a specific insurance, search whether it is in the vector of **insurances** or not, if yes, return true, otherwise, return false.

```
#ifndef DOCTOR_H_
#define DOCTOR_H_
#include "Person.h"
#include <iostream>
#include <vector>
using namespace std;
class Doctor : public Person
{
public:
    Doctor();
    bool takeInsurance(string myInsurance) const;

private:
    vector<string> insurances;
};
#endif
```

```
#include "Doctor.h"
```

```cpp
#include <iostream>
using namespace std;

Doctor::Doctor() : Person()
{
    insurances.push_back("medicaid");
    insurances.push_back("medicare");
}

bool Doctor::takeInsurance(string myInsurance) const
{
    for (int i = 0; i < insurances.size(); i++)
        if (insurances[i] == myInsurance)
           return true;

    return false;
}
```