Exam 2 Questions
CSE 232 (Introduction to Programming II)

1. If a string variable named `str` is declared, what is the type of the expression `&str`?

    (a) `string`
    (b) `string *`
    (c) `string &`
    (d) `& string`
    (e) (c) and (d)

2. You are given the following function:

```
void SomFunc(int a, int b){
 cout << a << " and " << b;
}
```

    What is the output of the following?

```
int a = 9;
int b = 3;
SomFunc(a++, --b);
```

    (a) 9 and 3
    (b) 9 and 2
    (c) 10 and 3
    (d) 10 and 2
    (e) Undefined behavior

3. Given the following function declarations, which of the following overloaded functions are valid?

```
void Green();
int Green();

int White(string);
int White(int);

string Sparty(string s);
string Sparty();
```

    (a) `Green` is overloaded
    (b) `White` is overloaded
    (c) `Sparty` is overloaded
    (d) (a) and (c)
    (e) (b) and (c)

4. In the example below, which statement does the `break` terminate?

```
for (...) { // For 1
 while (...) { // While 1
 }
 while (...) { // While 2
  switch (...) { // Switch 1
   case 0:
    ...
   default:
    for (...) { // For 2
     ...
    }
  }
 }
 break;
}
```

    (a) For 1
    (b) For 2
    (c) While 1
    (d) While 2
    (e) Switch 1

5. What is the difference between the following two loops?

```
for (int i = 15; i > 0; --i){
 ...
}
```

    And

```
int i = 15;
while (i > 0){
 ...
 i--;
}
```

    (a) The two loops are functionally identical
    (b) The values of the two `i`'s are different at the end of each loop iteration
    (c) The scope of `i` is different.
    (d) (b) and (c)

6. What is the correct way to declare a pointer to a char `c`?

    (a) `char *ptr = &c;`
    (b) `char &ptr = c;`
    (c) `char *ptr = c;`
    (d) (a) and (b)
    (e) (b) and (c)

7. Which of the following correctly declares a vector of doubles?

    (a) `vector<double> v;`
    (b) `double vector v;`
    (c) `vector v<double>;`
    (d) `double v<vector>;`
    (e) `vector double v;`

8. What is the output of the following?

```
vector<char> v = {'H', 'O', 'W',
'D', 'Y'};
for (int i = v.size() - 1; i > 0 ;
i--) {
 cout << v.at(i) << " ";
}
```

    (a) H O W D Y
    (b) Y D W O H
    (c) Y D W O
    (d) D W O H
    (e) D W O

9. What is the missing line in the code below, so that it outputs 7?

```
class MyClass {
 public:
   int x;
   MyClass(int val) : x(val) {}
};

int main() {
 MyClass obj1(7);
 // Missing code
 cout << obj2.x;
 return 0;
}
```

    (a) `MyClass obj2 = obj1;`
    (b) `MyClass &obj2 = obj1;`
    (c) `MyClass *obj2 = &obj1;`
    (d) (a) and (b)
    (e) (a) and (c)

10. You are given the following function:

```
void swap(int a, int &b) {
 int temp = a;
 a = b;
 b = temp;
}
```

What is the output of the following?

```
int x = 8, y = 17;
swap(x, y);
cout << x << " " << y;
```

    (a) 8 8
    (b) 8 17
    (c) 17 8
    (d) 17 17
    (e) Compile error

11. What is equivalent to `ptr->size();`?

    (a) `ptr.size();`
    (b) `*ptr->size();`
    (c) `*ptr.size();`
    (d) `(*ptr).size();`
    (e) (c) and (d)

12. Your program consists of the following files:

```
main.cpp
driver.cpp
driver.h
tests.cpp
tests.h
```

How would you complete the command `g++ -Wall -std=c++20` to compile your program?

(a) `main.cpp driver.cpp driver.h tests.cpp tests.h`
(b) `main.cpp driver.cpp tests.cpp`
(c) `main.cpp driver.h tests.h`
(d) `main.cpp`

13. What is the correct way of including a header file `myheader` that you wrote?

(a) `#include <myheader.h>`
(b) `#include "myheader.h"`
(c) `#include myheader.h`
(d) (a) and (b)
(e) (b) and (c)

14. What does the command `./a.out < a.txt > b.txt` do when executed at the command line?

(a) It redirects contents of `a.txt` to be the input for `a.out`, and redirects the output from `a.out` into `b.txt`
(b) It compares the exit code from `a.out` to the exit code from `a.txt` (less than) and `b.txt` (greater than)
(c) It compares the sizes of the files of `a.out` to `a.txt` (less than) and `b.txt` (greater than)
(d) It runs `a.out`, then runs `a.txt`, then runs `b.txt`.
(e) This is an invalid command

15. Which of the following is true about const functions?

(a) A const function does not change the object the function belongs to
(b) A const function can be called on a non-const object
(c) A const function can be called on a const object
(d) (a) and (c)
(e) All of the choices

16. Which of the following is true about getter functions?

(a) They usually have a return type
(b) They are usually void functions
(c) They are usually const functions
(d) They are usually non-const functions
(e) (a) and (c)
(f) (b) and (d)

17. Which of the following is true?

(a) `operator<<` works with `istream`
(b) `operator>>` works with `ostream`
(c) `operator<<` works with `ostream`
(d) `operator>>` works with `istream`
(e) (a) and (b)
(f) (c) and (d)

18. When should `fstream` be used?

(a) Reading from and writing to the terminal
(b) Reading from and writing to files
(c) IO redirection
(d) All of the choices

19. Which of the following git commands can be executed from the command line to make the current working directory a git repository?

(a) `git add`
(b) `git commit`
(c) `git config`
(d) `git init`
(e) `git cd`

20. Which of the following git commands is used to download commits from a remote repository into an existing local repository?

 (a) `git clone`
 (b) `git pull`
 (c) `git innit`
 (d) (a) and (b)
 (e) All of the choices

21. Which of the following statements is true?

 (a) A local repository can exist without a remote repository
 (b) A remote repository can exist without a local repository
 (c) Both statements are true
 (d) None of the statements are true

22. Which of the following is an example of a valid `HTTPS URL` of a repository?

 (a) `https://github.com/user/project.git`
 (b) `git@github.com:user/project.git`
 (c) `https://github.com/user/project`
 (d) (a) and (b)
 (e) (b) and (c)

23. What is type of `x` in the following code?

```
const char* c = "Engine";
auto x = c[1];
```

 (a) `char`
 (b) `const char`
 (c) `string`
 (d) `const string`
 (e) `"Engine"` cannot be stored this way

24. Which of the following is an example of a multi-line comment in `C++`?

 (a) `\\Commented text`
 (b) `// Commented text`
 (c) `<!-- Commented text -->`
 (d) `\* Commented text *\`
 (e) `/* Commented text */`

25. If `r` is a reference, how do you access its value?

 (a) `r`
 (b) `&r`
 (c) `*r`
 (d) You cannot directly access `r`

26. How many arguments does a default constructor take in?

 (a) No argument
 (b) One argument
 (c) At least one argument
 (d) Depends on the class

27. How many constructors can a class have?

 (a) None
 (b) Exactly one
 (c) One or more
 (d) At most one

28. You have a function that takes a vector of integers `v` and an index `i` as function parameters, and returns the value stored in `v` at the index `i`. Which of the following exceptions should you check for (and throw) in this function?

 (a) `logic_error`
 (b) `invalid_argument`
 (c) `domain_error`
 (d) `length_error`
 (e) `out_of_range`

29. Which of the following is an advantage of using assertions over exceptions?

 (a) Assertions can be easily turned off
 (b) Assertions can be used in compile-time, unlike exceptions
 (c) Assertions are more optimized than exceptions
 (d) All of the choices
 (e) None of the choices

30. Which of the following commented lines can `var` not be accessed from?

```
for (;;) {
 string var;
 // Label A
 while (x.length() < 3) {
  // Label B
 }
 // Label C
}
// Label D
```

   (a) Label A
   (b) Label B
   (c) Label C
   (d) Label D
   (e) It can be accessed from everywhere

31. If the following line of code does not cause a compiler error, which of the following is a valid type for `x`?

```
const int y = x[0];
```

   (a) `const vector<int>`
   (b) `vector <const int>`
   (c) `vector <double>`
   (d) All of the choices

32. Consider the following code:

```
const MyClass someObj;
```

Which of the following member functions cannot be called on `someObj`, assuming the functions do exactly as their names imply?

   (a) `getValue`
   (b) `setValue`
   (c) `printValue`
   (d) (a) and (b)
   (e) None of these functions can be called

33. What makes a `const MyClass *` different from a `MyClass *`?

   (a) A `const MyClass *` cannot change the object it points to
   (b) A `const MyClass *` cannot change to point at a different object
   (c) They are identical
   (d) A `const MyClass *` must be initialized, not just declared

34. Which of the following is true about assertions?

   (a) They can generate run-time errors
   (b) They can generate compile-time errors
   (c) They are part of the `<assert>` library
   (d) None of the choices

35. Which git command is used to send a local repository's commits to a remote repository?

   (a) `git add`
   (b) `git commit`
   (c) `git fetch`
   (d) `git pull`
   (e) `git push`

36. A const member function has what property that distinguishes it from a non-const member function?

   (a) It can be called on const objects
   (b) It has only const parameters
   (c) It returns a const object
   (d) It returns a reference to a const object
   (e) All of the choices

37. What is the purpose of header guards?

   (a) To allow for faster compilation
   (b) To allow templates to be instantiated
   (c) To avoid redeclaration/redefinition errors
   (d) To ensure that a class's privacy is maintained
   (e) All of the choices

38. You have a class `MyClass` with a member function `run` that does not take any function parameters. Now, what is the missing line in the code below that calls the function `run`?

```
MyClass o;
//missing code
```

   (a) `run();`
   (b) `MyClass.run;`
   (c) `MyClass.run();`
   (d) `o.run;`
   (e) `o.run();`

39. In the code below, what is the initial value of `i` in debugging mode if the breakpoint is set at the line of the comment?

```
for(int i = 10; i >= 5; i--){
 cout << i; //Breakpoint here
}
```

   (a) 0
   (b) 5
   (c) 9
   (d) 10
   (e) Undefined/Garbage

40. What is the missing line in the code below, so that the output is `HELLO`?

```
const char* c = "HELLO";
 int i = 0;
 //Missing code
 {
  cout << c[i];
  i++;
 }
```

   (a) `while(i < c.size())`
   (b) `while(i < size(c))`
   (c) `while(c[i] != '\0')`
   (d) `while(c[i] != '/0')`