This example exam is mainly to demonstrate the format of the exam. The actual exam will likely be a bit more diverse. Practice using both arrays on the heap and STL data structures because the actual exam may be set up so that one is more feasible than the other.

Moving to skel:
There is a Makefile in each folder. You must start in the root folder and enter
`make skel`
This will move the entire folder structure to your skel home directory, under ***cpp_exampleexam***.
In each problem folder **`make`** will build your program while **`make skel`** will move the relevant files from the directory on your computer into the same directory on skel.
If **`make`** does not work, copy the commands from ***Makefile*** and use those directly.

Part A: Knowledge (20%)
Multiple choice questions in separate quiz assignment on Canvas

Part B: Basic core skills (30%)

1. Make the operation ***add***
    - Takes two integers as parameters
    - Returns the sum of those two integers
    - ***main()*** in folder B1 should compile and run on skel.ru.is
2. Make the class ***Adder***
    - Has a constructor which takes one integer as a parameter
    - Has one operation ***add***
        - Takes one integer as parameter
        - Returns the sum of the parameter and the original constructor parameter
    - ***main()*** in folder B2 should compile and run on skel.ru.is
3. Make the operation ***read_file_and_add()***
    - The operation first asks the user to input a filename on the terminal
        - The user should have .txt at the end of the input
        - Use the file ***example_input.txt*** for testing
    - It then opens the file
        - Each line is <some_integer> <some_integer>
        - Example:
            - **`12 89`**
              `34 21`
              `1 7`
        - For each line in the file print to the terminal
            - The sum of <some_int> and <some_int> is <sum_result>
        - Example:
            - **`The sum of 12 and 89 is 101`**
    - ***main()*** in folder B3 should compile and run on skel.ru.is

Part C: Core skills (30%)

- Make the operation ***add***
  - Takes two parameters of the same type, which can be any class or type
  - Returns the sum of those two parameters as determined by the operator +
- Make the class ***Addable***
  - Has a constructor which takes an integer and a double as a parameter
  - Overloads the ***operator +***
    - Is applied to two instances/references of ***Addable*** (operands)
    - Returns an instance of ***Addable***
      - The integer is the sum of the integers from both operands
      - The real number is the sum of the doubles from both operands
  - Overloads the ***operator <<***
    - Is applied to an ostream and an instance of ***Addable***
    - Sends the classes information (with simple labels) to the stream
    - Returns the stream
- *The following 4 operations in the class Addable don't need an instance to be called*:
- Make the operation ***add_and_return***
  - Also takes an integer as a parameter
    - Determines how many pairs of ***Addable*** should be made
  - Make that many pairs of ***Addable*** with random values
  - For each pair call ***add(operand1, operand2)*** and save the result
  - Returns collection of Addable
    - You can determine type of collection
- Make the operation ***print_addables_to_terminal***
  - Takes a collection of addable as a parameter
    - Same type of collection as returned from ***add_and_return***
  - print all instances to the terminal
- Make the operation ***write_addables_to_file***
  - Takes a filename as a parameter
    - Opens/creates this file as a binary file for writing
  - Also takes a collection of addable as a parameter
    - Same type of collection as returned from ***add_and_return***
  - Write all the instances (oper,oper,result,oper,oper,result,...) to a binary file
- Make the operation ***read_addables_from_file***
  - Takes a filename as a parameter
    - Opens this file as a binary file for reading
  - Reads from the same file format as was written in the previous problem
    - Decided by yourself
  - Returns collection of Addable
    - Same type of collection as returned from ***add_and_return***
- ***main()*** in folder C should compile and run on skel.ru.is

Part D: Advanced skills (20%)
Following are four problems.
Solve one of them correctly for 15%
Solve another one of them correctly for 5%

No specific examples here. These problems can be anything.