**Large Size Object**

**Writer:** Sayak Haldar

Here is a listing of C++ quiz on “Large Objects” along with answers, explanations and/or solutions:

**1. How to store the large objects in c++ if it extents its allocated memory?**

a) memory heap

b) stack

c) queue

d) none of the mentioned

**Answer: a**

**2. When we are using heap operations what do we need to do to save the memory?**

a) rename the objects

b) delete the objects after processing

c) both rename & delete the objects

d) none of the mentioned

**Answer: b**

Explanation: when you allocate memory from the heap, you must remember to clean up objects when you’re done! Failure to do so is called a memory leak.

**3. Which container in c++ will take large objects?**

a) string

b) class

c) vector

d) none of the mentioned

**Answer: c**

Explanation: Because vector is mainly used to store large objects for game

programming and other operations etc.

**4. What is the output of this program?**

#include <iostream>

using namespace std;

class sample

{

public:

sample()

{

cout << "X::X()" << endl;

}

sample( sample const & )

{

cout << "X::X( X const & )" << endl;

}

sample& operator=( sample const & )

{

cout << "X::operator=(X const &)" << endl;

}

};

sample f()

{

sample tmp;

return tmp;

}

int main()

{

sample x = f();

return 0;

}

a) X::operator=(X const &)

b) X::X( X const & )

c) X::X()

d) None of the mentioned

**Answer: c**

Explanation: As we are passing the object without any attributes it will return as X::X().

Output:

$ g++ large.cpp

$ a.out

X::X()

**5. How to stop your program from eating so much ram?**

a) Find a way to work with the data one at a time

b) Declare it in program memory, instead of on the stack

c) Use the hard drive, instead of RAM

d) All of the mentioned

Answer: d

**6. Which option is best to eliminate the memory problem?**

a) use smart pointers

b) use raw pointers

c) use virtual destructor

d) use smart pointers & virtual destructor

**Answer: d**

Explanation: Virtual destructor is one essential way to prevent memory leak when using Base class pointer we create a derived class object dynamically. It forces the object to be destructed in reverse order in which it was constructed and the smart pointer will delete the object from memory when the object goes out of scope.

**7. What is the size of the heap?**

a) 10MB

b) 500MB

c) 1GB

d) Size of the heap memory is limited by the size of the RAM and the swap memory

Answer: d

Explanation: None.

**8. How to unlimit the size of the stack?**

a) setrlimit()

b) unlimit()

c) both setrlimit() & unlimit()

d) none of the mentioned

**Answer: a**

Explanation: None.

**9. In linux, how does the heaps and stacks are managed?**

a) ram

b) secondary memory

c) virtual memory

d) none of the mentioned

**Answer: c**

Explanation: In virtual memory, We can keep track of all the objects and access them much faster than any another. (Is not virtual memory: the pages, page index tables etc are itself part of the RAM? For faster access, frequently accessed variables are cached)

**10. Which is used to pass the large objects in c++?**

a) pass by value

b) pass by reference

c) both pass by value & reference

d) none of the mentioned

**Answer: b**

Explanation: Because by using pass by reference we need to pass only address location, So it can save a lot of memory.