Vector – MCQs

1. Choose the correct option or options about vectors
   1. Used to store elements of similar data type.
   2. The size of a vector can be changed at runtime.
   3. The vectors are part of the C++ Standard Template Library.
   4. All of the above
2. Choose the correct way to initialize a vector
   1. vector<int> \_vector = {10, 20, 30 ,40 50, 60, 70, 80, 90, 100};
   2. vector<int> \_vector {10, 20, 30 ,40 50, 60, 70, 80, 90, 100};
   3. vector<int> \_vector(10, 0);
   4. All of the above
3. The basic vector operation/s
   1. Add elements
   2. Add & Access elements
   3. Add, Access & Change elements
   4. Add, Access, Change & Remove elements
4. How would you add more elements in a vector<int> \_vector = {101, 102, 103, 104}; ?
   1. \_vector.push\_front(105);
   2. \_vector.push\_back(105);
   3. \_vector.push\_front(105, 106);
   4. \_vector.push\_back(105, 106);
5. \_\_\_\_\_\_ can be primitive as well as user-defined.
   1. Array
   2. Vector
   3. Both
   4. None
6. Where would you find the lower\_bound() and upper\_bound() methods for vector?
   1. #include<iostream>
   2. #include<vector>
   3. #include<algorithm>
   4. All of the above
7. A \_\_\_\_\_\_ implements an array internally but this array is free to add elements even beyond its predefined size.
   1. Array
   2. Dynamic Array
   3. Vector
   4. All of the above
8. How would you add a value 500 in the vector<int> \_vector = {100, 200, 300, 400} at position 10?
   1. \_vector.insert(\_vector.begin() + (\_vector.size()), 500);
   2. \_vector.insert(\_vector.begin() + (\_vector.size()+5), 500);
   3. \_vector.insert(\_vector.begin() + (\_vector.size()++), 500);
   4. \_vector.insert(\_vector.begin() + (++\_vector.size()), 500);
9. What are the available methods to copy a vector vector<int> \_vector = {10, 20, 30, 40, 50};
   1. vector<int> v = \_vector; and vector<int> v(\_vector);
   2. vector<int> v;

copy(\_vector.begin(), \_vector.end(), back\_inserter(v));

* 1. vector<int> v;

v.assign(\_vector.begin(), \_vector.end());

* 1. All of the above

1. What method would you use to modify a vector vector<int> \_vector = {10, 20, 30, 40, 50} to 10,100,30,1000,50?
   1. \_vector.set(1) = 100; and \_vector.set(3) = 1000;
   2. \_vector.update(1) = 100; and \_vector.update(3) = 1000;
   3. \_vector.at(1) = 100; and \_vector.at(3) = 1000;
   4. \_vector.insert(1) = 100; and \_vector.insert(3) = 1000;
2. What is the output of this code snippet below

#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

class VectorRule {

vector<int> axisX;

vector<int> axisY(5,0);

public:

VectorRule (vector<int> axisX){

cout<<endl<<"\n\tGenerate the Axis-Y:";

this->axisX = axisX;

}

Void getVectorRule(){

cout<<"\n\tX"<<"\tY";

for(int i=0; i<axisX.size(); i++){

cout<<"\n\t"<<axisX<<"\t"<<axisY;

}

}

};

int main(){

vector<int> v1 = {10, 20, 30, 40, 50};

vector<int> \_vector(v1);

VectorRule obj(\_vector);

obj.getVectorRule();

return 0;

}

* 1. Compile-Time Error
  2. Run-Time Error
  3. No problem and runs well
  4. None

1. What would be the output of the below code snippet,

int main(){

vector<int> vectorCollection;

for (int i = 1; i <= 10; i++)

vectorCollection.push\_back(i \* 10);

int\* pos = vectorCollection.data();

cout << \*pos;

return 0;

}

* 1. 0
  2. 10
  3. 100
  4. 102030405060708090100