**North South University - Spring 2023**

Course: CSE225L Assessment: Lab 1 / HW 1 / PS 1

Section: 06 NSU ID: 2211424642 Name: Joy Kumar Ghosh

// Task-1 Codes

|  |
| --- |
| template <class T>  class dynArr{  private:  T \*data;  int size;  public:  dynArr();  dynArr(int);  ~dynArr();  void setValue(int, T);  T getValue(int);  }; |
| template <class T>  class dynArr{  private:  T \*data;  int size;  public:  dynArr();  dynArr(int);  ~dynArr();  void setValue(int, T);  T getValue(int);  }; |
| int main()  {  dynArr<int> obj1;  dynArr<double> obj2(5);  double num;  for(int i = 0; i < 5; i++){  cout << "Enter " << (i+1) << "-th number: ";  cin >> num;  obj2.setValue(i, num);  }  cout << endl << endl << "Elements of 2nd Object: " << endl;  for(int i = 0; i < 5; i++){  cout << obj2.getValue(i) << " ";  }  cout << endl;  return 0;  } |
|  |

// Task-2 Codes

|  |
| --- |
| template <class T>  class dynArr{  private:  T \*data;  int size;  public:  dynArr();  dynArr(int);  ~dynArr();  void setValue(int, T);  T getValue(int);  void allocate(int s);  }; |
| template <class T>  dynArr<T>::dynArr(){  data = NULL;  size = 0;  }  template <class T>  dynArr<T>::dynArr(int s){  data = new T[s];  size = s;  }  template <class T>  dynArr<T>::~dynArr(){  delete [] data;  }  template <class T>  T dynArr<T>::getValue(int index){  return data[index];  }  template <class T>  void dynArr<T>::setValue(int index, T value){  data[index] = value;  }  template <class T>  void dynArr<T>::allocate(int s){  delete[] data;  data = new T[s];  size = s;  } |
| int main()  {  dynArr<int> obj1;  dynArr<double> obj2(5);  double num;  cout << "Enter elements for obj2: " << endl;  for(int i = 0; i < 5; i++){  cout << "Enter " << (i+1) << "-th number: ";  cin >> num;  obj2.setValue(i, num);  }  cout << endl << endl << "Elements of 2nd Object: " << endl;  for(int i = 0; i < 5; i++){  cout << obj2.getValue(i) << " ";  }  cout << endl;  obj1.allocate(5);  obj2.allocate(10);  cout << endl << endl << "Enter elements for obj1: " << endl;  for(int i = 0; i < 5; i++){  cout << "Enter " << (i+1) << "-th number: ";  cin >> num;  obj1.setValue(i, num);  }  cout << endl << endl << "Elements of 1st Object: " << endl;  for(int i = 0; i < 5; i++){  cout << obj1.getValue(i) << " ";  }  return 0;  } |
|  |

// Task-3 Codes

|  |
| --- |
| template <class T>  class dynArr{  private:  T \*data;  int row;  int column;  public:  dynArr();  dynArr(int, int);  ~dynArr();  void setValue(int, int, T);  T getValue(int, int);  void allocate(int, int);  }; |
| template <class T>  dynArr<T>::dynArr(){  data = NULL;  row = 0;  column = 0;  }  template <class T>  dynArr<T>::dynArr(int r, int c){  data = new T[r\*c];  row = r;  column = c;  }  template <class T>  dynArr<T>::~dynArr(){  delete [] data;  }  template <class T>  T dynArr<T>::getValue(int r, int c){  return \*(data + r \* column + c);  }  template <class T>  void dynArr<T>::setValue(int r, int c, T value){  \*(data + r \* column + c) = value;  }  template <class T>  void dynArr<T>::allocate(int r, int c){  delete[] data;  data = new T[r\*c];  row = r;  column = c;  } |
| int main()  {  int row, column, temp;  cout << "Enter row size: ";  cin >> row;  cout << "Enter column size: ";  cin >> column;  //declaration with integer type  dynArr<int> object1(row, column);  cout << endl << "Enter all elements(" << row << ", " << column << "): " << endl;  for(int i = 0; i < row; i++){  for(int j = 0; j < column; j++){  cout << "Enter elements of (" << (i+1) << ", " << (j+1) << "): ";  cin >> temp;  object1.setValue(i, j, temp);  }  }  cout << endl << "All elements in matrix formate: " << endl;  for(int i = 0; i < row; i++){  for(int j = 0; j < column; j++){  cout << object1.getValue(i, j) << " ";  }  cout << endl;  }  object1.~dynArr();  cout << endl;  return 0;  } |
|  |

// Task-4 Codes

|  |
| --- |
|  |
|  |
|  |

// Task-5 Codes

|  |
| --- |
|  |
|  |
|  |