Practical 1

1. Design an item class for reading and displaying the item details information, the getdata() and putdata() methods will be used respectively. Where all methods will be public method.

Code:

|  |
| --- |
| #include<iostream>  // class to store items  class items  {  private:      // no. of items in inventory      int number;      // price of each item      float cost;  public:      // function to record data      void getData(int x, float y)      {          number = x;          cost = y;      }      // function prototype      void printData(void);  };  // function to print data  void items::printData(void)  {      std::cout << "the number of items is: " << number <<"\n";      std::cout << "the cost of each item is: " << cost <<"\n";  }  int main()  {      // initializing first item      items mobile;      std::cout << "\nObject one >>\n";      // add data to inventory      mobile.getData(10011, 2899.95);      // print the current data in inventory      mobile.printData();      // initializing second item      items computer;      std::cout << "\nObject Two >>\n";      // add data to inventory      computer.getData(20011, 1575.50);      // print the current data in inventory      computer.printData();      return 0;  } |

Output:

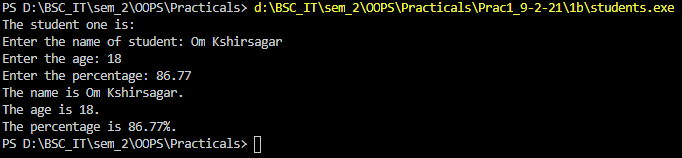


1. Design the class student containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively. Where getData() will be private method.

Code:

|  |
| --- |
| #include<iostream>  #include<string>  using namespace std;  // class to manage students  class student  {  private:      // name of student      string name;      // age of student      int age;      // student's percentage      float percentage;      // function to record data      void getData()      {          cout << "Enter the name of student: ";          getline(cin, name);          cout << "Enter the age: ";          cin >> age;          cout << "Enter the percentage: ";          cin >> percentage;      }  public:      // fucntion to print data to console      void displayData()          {              // get the name, sage, percentage from user              getData();              std::cout << "The name is " << name <<".\n";              std::cout << "The age is " << age <<".\n";              std::cout << "The percentage is " << percentage <<"%.\n";          }  };  int main()  {      // initialize a student      student John;      cout << "The student one is: \n";      // record and display data      John.displayData();      return 0;  } |

Output:

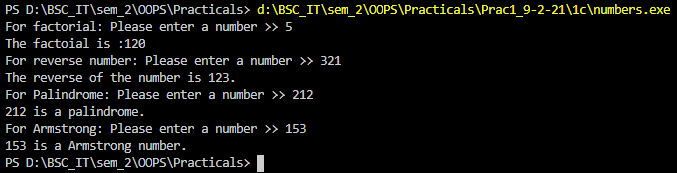


1. Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not. Where readNo() will be private method.

Code:

|  |
| --- |
| #include<iostream>  #include<cmath>  using namespace std;  // a class to store all the numbers and perform actions on them  class numbers  {  private:      // variable to store the number      int number;      // function to get number and store in the number variable      void readno()      {          // get input          cout << "Please enter a number >> ";          cin >> number;      }  public:      // method to calculate factorial      void factorial();      // method to find reverse of a number      int reverseNo();      // method to check if a number is palindrome or not      void isPalindrome();      // method to check if a number is armstrong or not      void isArmstrong();  };  int main()  {      // create a class member      numbers first;      // calculate factorial      first.factorial();      // calculate reverse of number      cout << "For reverse number: ";      int revNum = first.reverseNo();      cout << "The reverse of the number is " << revNum << ".\n";      // check and print if a number is palindrome or not      first.isPalindrome();      // check and print if a number is armstrong or not      first.isArmstrong();      return 0;  }  void numbers::factorial()  {      cout << "For factorial: ";      // get the number      readno();      // create a copy of the number and initialize the required variables      int temp = number, ans = 1;      // calculate      while(temp > 1)      {          ans \*= temp;          temp--;      }      // print      cout << "The factoial is :" << ans <<"\n";  }  int numbers::reverseNo()  {      // get the number      readno();      // create a copy of the number and initialize the required variables      int num = number, temp, revNum = 0;      // calculate      while(num > 0)      {          temp = num % 10;          revNum = revNum \* 10 + temp;          num /= 10;      }      /\*  return the number instead of printing as          we need to calculate reverse for palindrome\*/      return revNum;  }  void numbers::isPalindrome()  {      cout << "For Palindrome: ";      // get the number and calculate reverse and store the number in variable      int revNum = reverseNo();      // check if it is palindrome or not      if (revNum == number)          cout << revNum << " is a palindrome.\n";      else          cout << number << " is not a palindrome.\n";  }  void numbers::isArmstrong()  {      cout << "For Armstrong: ";      // get number      readno();      // create a copy of the number and initialize the required variables      int num = number, digit, sum = 0;      // calculate      while(num != 0)      {          digit = num % 10;          sum += pow(digit, 3);          num /= 10;      }      // check if it is armstrong or not      if (sum == number)          cout << sum << " is a Armstrong number.";      else          cout << number << " is not a Armstrong number.";  } |

Output:



1. Aim: Write a program to demonstrate function definition outside class and accessing class members in function definition.

Code:

/\*

    Write a program to demonstrate function

    definition outside class and accessing

    class members in function definition.

\*/

#include<iostream>

class numbers

{

private:

    int num1, num2;

public:

    numbers(int x, int y);

    numbers(int x);

    int addMemb();

    ~numbers();

};

int main()

{

    int i, j;

    std::cout << "Enter two numbers :";

    std::cin >> i >> j;

    numbers A(i, j);

    std::cout << "The addition of "<<i << " " << j <<" is :"  << A.addMemb()<<std::endl;

}

numbers::numbers(int x, int y)

{

    num1 = x;

    num2 = y;

}

int numbers::addMemb()

{

    return num1+num2;

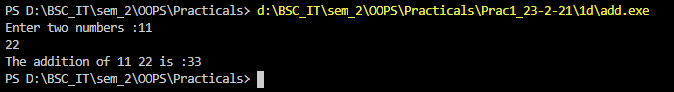
}

numbers::~numbers()

{

}

Output:



Write-Up:

