东 南 大 学 考 试 卷（ B 卷）

学号 姓名

封

线

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | 程序设计基础及语言II（双语） | | 考试学期 | 19-20-3 |  |
| 适用专业 | 计算机大类 | 考试形式 | 半开卷  可带一本教材 | 考试时间长度 | 笔试  60分钟 |
| 机试  120分钟 |

总分：**100分**

|  |  |  |
| --- | --- | --- |
| **试题** | **得分** | **评阅人** |
| 第一部分：笔试 |  |  |
| 第二部分：机试 |  |  |
| 成绩合计： |  |  |

东 南 大 学 考 试 卷（ B 卷）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 程序设计基础及语言II（双语） | | 考试学期 | | 19-20-3 | | 卷面 | 40分 | |
| 适用专业  学号 姓名  密  封  线 | 计算机大类 | 考试形式 | | 半开卷  可带一本教材 | | 考试时间长度 | | | 笔试  60分钟 |

仅 允 许 携 带 课 程 指 定 教 材

请将答案写在答题纸上

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一部分 笔试  Note: All answers must be written on answer sheet!  I Read the programs as follows, and please write down the output of them：（20 score）  1．What is the output of the following? (4 score)  class A {  private:  int x;  static int y;  };  class B : public A {  private:  int x, z;  };  class C : public A {  private:  int x, y;  A m;  };  int main()  {  cout<<sizeof(A)<<endl<<sizeof(B)<<endl<<sizeof(C)<<endl;  }  2．What is the output of the following? (4 score)   |  |  | | --- | --- | | #include <iostream>  using namespace std;  class Shape {  static int number;  public:  Shape() { number++; }  ~Shape() { number--; }  static void print()  {  cout<< number<<endl;  }  };  int Shape::number = 0;  class Circle :public Shape {  public:  Circle(int r) { radius = r; }  int radius;  }; | Shape \* s1 = new Shape();  void main()  {  Shape::print() ;  Shape s2;  Circle \* c1 = new Circle(1);  {  Circle c2(2);  Shape::print();  }  delete c1;  Shape::print();  {  static Circle c3(3);  }  Shape::print();  } |   3．What is the output of the following? (4 score)  #include <iostream>  using namespace std;  template<class T,int N>  class sum  {  public:  T\* f(T a[],T b[]) ;  };  template<class T, int N>  T\* sum<T,N>::f(T a[],T b[])  {  T\* c= new T[N];  for(int i=0;i<N;i++)  c[i]=a[i]+b[i];  return c;  }  void main()  {  sum<int,5> temp;  int a[2][5] = {1,2,3,4,5,10,20,30,40},\*p;  p=temp.f(a[0], a[1]);  for(int i=0;i<5;i++)  cout<<p[i]<<endl;  }  4．What is the output of the following? (4 score)  class A  {  public:  void f1() { cout << "A.f1()" << endl; }  virtual void f2() { cout << "A.f2()" << endl; f1(); }  virtual void f3() { cout << "A.f3()" << endl; f2(); }  virtual ~A() { cout << "A dtor called" << endl; }  };  class B : public A  {  public:  virtual void f1() { cout << "B.f1()" << endl; }  void f2() { cout << "B.f2()" << endl; }  virtual ~B() { cout << "B dtor called" << endl; }  };  int main()  {  A \*br = new B();  br->f1(); br->f2(); br->f3();  delete br;  }  5．What is the output of the following? (4 score)   |  |  | | --- | --- | | #include <iostream>  using namespace std;  class MyExcept :public runtime\_error  {  public:  MyExcept() : runtime\_error("Except: MyExcept"){}  MyExcept(const MyExcept &)  : runtime\_error("Except: MyExcept")  {  cout << "Copy constructor \n";  }  ~MyExcept() {}  };  class Sample  {  public:  Sample( ) {  cout << "Constructor of Sample\n";  }  ~Sample( ) {  cout << "Deconstruct of Sample\n";  }  }; | void MyFun()  {  Sample s;  cout << "Before throwing \n";  throw MyExcept();  cout << "After throwing \n";  }  void main()  {  try {  MyFun();  }  catch (MyExcept &e)  {  cout << e.what() << endl;  }  catch (...)  {  cout << "Catching others\n " ;  }  } |   II．To fill in the following blanks to complete program segments. （20 scores）   1. （10 scores）Fill in the blanks to complete the member functions of String class according to the comments.   String::String( (1 ) ) : length(copy.length) // copy constructor  {  cout << "Copy constructor: " << copy.sPtr << endl;  setString(copy.sPtr);  }  const String &String::operator=(const String &right) // overloaded = operator  {  cout << "operator= called" << endl;  if ( (2) ) // avoid self assignment  {  delete[] sPtr; // prevents memory leak  length = right.length; // new String length  setString(right.sPtr); // call utility function  } // end if  else  cout << "Attempted assignment of a String to itself" << endl;  return \*this; // enables cascaded assignments  }  // Is this String equal to right String?  bool String::operator==(const String &right) const  {  return (3) ;  }  // return reference to character in String as a modifiable lvalue  (4) String::operator[](int subscript)  {  if (subscript < 0 || subscript >= length)  {  cerr << "Error: Subscript " << subscript  << " out of range" << endl;  exit(1); // terminate program  } // end if  return sPtr[subscript]; // non-const return; modifiable lvalue  }  (5) operator<<(ostream &output, const String &s)  {  output << s.sPtr;  return output; // enables cascading  }   1. Filling the blanks to make the program successful. (10scores)   #include <iostream>  using namespace std;  class Point  {  public:  Point(double x = 0, double y = 0) :X(x), Y(y) {}  double getX()const { return X; }  double getY()const { return Y; }  (1) double area()const (2) ;  virtual void print()  {  cout << "Center=[" << X << "," << Y << "]";  }  protected:  double X, Y;  };  class Circle :public Point  {  friend \_\_\_\_(3) (ostream&, const Circle&);  public:  Circle(double x = 0, double y = 0, double r = 0) :Point(x, y), radius(r) {}  double getRadius()const { return radius; }  double area()const  {  return 3.14159\*getRadius()\*getRadius();  }  void print()  {  Point::print();  cout << ", Radius = " << radius;  // cout << ", area=" << area() << endl << endl;  }  private:  double radius;  };  class Cylinder :public Circle  {  friend (3) (ostream&, const Cylinder&);  public:  Cylinder(double x = 0, double y = 0, double r = 0, double h = 0) : Circle(x, y, r), height(h) {}  double getHeight()const { return height; }  double area()const  {  return 2 \* Circle::area() + 2 \* 3.14159\*getRadius()\*getHeight();  }  double volume()const  {  return Circle::area()\*height;  }  void print()  {  Circle::print();  cout << ", Height = " << height;  cout << ", area =" << area() << endl;  cout << ", volume =" << volume() << endl << endl;  }  protected:  (4)  };  (3) (ostream &output, const Circle& c)  {  output << "Center=[" << c.X << "," << c.Y << "], r=" << c.getRadius() << ", area=" << c.area() << endl;  return output;  }  (3) (ostream &output, const Cylinder& cy)  {  output << "Center=[" << cy.X << "," << cy.Y << "], r=" << cy.getRadius() << ", h=" << cy.height << ", area=" << cy.area() << ", volume=" << cy.volume() << endl;  return output;  }  int main()  {  Point \* p[4];  Circle c1(3.5, 6.4, 5.2);  Circle c2(5.0, 4.0, 6.0);  Cylinder cy1(3.5, 6.4, 5.2, 10.2);  Cylinder cy2(5, 4, 6, 7.3);  p[0] = &c1;  p[1] = &c2;  p[2] = &cy1;  p[3] = &cy2;  for (int i = 0; i < 4; i++)  {  cout << "The basic information is:" << endl;  (5) print();  cout << endl;  cout << "The area is:" << (5) area()<<endl;  cout << endl;  }  cout << endl;  cout << "重载输出运算符的结果" << endl;  cout << cy1 << cy2 << endl;  }  东 南 大 学 考 试 卷（ B 卷）   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 课程名称 | 程序设计基础及语言II（双语） | | 考试学期 | | 18-19-3 | | 卷面 | 60分 | | | 适用专业 | 计算机大类 | 考试形式 | | 半开卷  可带一本教材 | | 考试时间长度 | | | 机试  120分钟 |   仅 允 许 携 带 课 程 指 定 教 材  第二部分 机试  要求：   * 1. 在本地D：或E：盘中，建立自己的文件夹，用来完成程序的编写和调试。   2. 建议：第一题Project(项目名)为Pro1，第二题Project(项目名)为Pro2，以此类推。   III. To write and test programs according to the requirements（60 scores）  1. **(20 scores)**According to the following pictures, implement the two classes (**A** and **B**) and test them. Assume that every member function only output its function name (except member function fa3 of class A). (By separating interface from implementation)    **2.(20 scores)** Create a file each line containing username and password, and then design a program to manage the username and password for each employee of the company. The following operations are required:  1) Add new employee and set password. The input format is: “Add *name* *password*”, where name and password are strings; If the employee already exists, output "Exist!"  2) Change the password. The input format is “Modify *name* *password*”. If the employee exists, modify his password. Otherwise, output "Not exist! "  3) Enter the company. The input format is: “Enter *name* *password*”. If the name and password are recorded, output "Yes". Otherwise, output "No".  When the following operations are entered:  Add *Jack 123w*  Add *Tom 23dk*  Modify *Marry 23lfd*  Enter *Jack 234*  Enter *Tom 23dk*  Add *Jack 2334*  The output should be:  Not exist!  No  Yes  Exist!  3. **(20 scores)**Complete the definition of class **ForwardList** to make main function output as the follows.   * 1. Need to implement a member function **FindKthFromTail** that outputs the value of the kth node from the bottom of the linked list.   2. Cannot modify the data members of class **ForwardList** and class **Node.**  |  |  | | --- | --- | | #include <iostream>  using namespace std;  template< typename T > class ForwardList;  template< typename T >  class Node  {  friend class ForwardList< T >;  public:  Node(const T &info) : data(info), nextPtr(NULL) {}    private:  T data; // data  Node< T > \*nextPtr; // next node in list  };  template< typename T >  class ForwardList  {  public:    ForwardList(): headPtr(NULL){}  ~ForwardList()  {  //finish the function implementation  }  // insert node at front of list  void insertAtFront(const T &value)  {  //finish the function implementation  }    void FindKthFromTail(int k)  {  //finish the function implementation  }  // is List empty?  bool isEmpty() const  {  return headPtr == NULL;  } // end function isEmpty  // display contents of List  void print() const  {  if (isEmpty()) // List is empty  {  cout << "The list is empty\n\n";  return;  } // end if  Node< T > \*currentPtr = headPtr;  cout << "The list is: ";  while (currentPtr != NULL) // get element data  {  cout << currentPtr->data << ' ';  currentPtr = currentPtr->nextPtr;  } // end while  cout << endl;  } // end function print  private:  Node< T > \*headPtr; // pointer to first node  }; // end class List | void main()  {  ForwardList<int> list;  list.insertAtFront(1);  list.insertAtFront(2);  list.insertAtFront(3);  list.insertAtFront(4);  list.insertAtFront(5);  list.insertAtFront(6);  list.insertAtFront(7);  list.insertAtFront(8);  list.insertAtFront(9);  list.insertAtFront(10);  list.print();  cout << endl;  list.FindKthFromTail(-1);  cout << endl;  list.FindKthFromTail(5);  cout << endl;  list.FindKthFromTail(15);  cout << endl ;  } | | Output：  This list is：10 9 8 7 6 5 4 3 2 1  K值输入小于等于0  该链表的倒数第k个值为：5  K输入大于链表长度  Destroying nodes …  10  9  8  7  6  5  4  3  2  1  All nodes destroyed | |   东 南 大 学 考 试 答 题 纸　　ＡＮＳＷＥＲ　ＳＨＥＥＴ   |  | | --- | | 参考答案：  阅读1  4  12  16  阅读2  1  4  2  3  阅读3  11  22  33  44  5  阅读4  A.f1()  B.f2()  A.f3()  B.f2()  B dtor called  A dtor called  阅读5  Constructor of Sample  Before throwing  Deconstruct of Sample  Except: MyExcept  填空1   1. const String &copy 2. &right != this 3. strcmp(sPtr, right.sPtr) == 0 4. char& 5. ostream&   填空2   1. Virtual 2. = 0 3. ostream& operator <<③ 4. double height; 5. p[i]->或者(\*(p+i))-> | |