**Lab Task-9**

**Instructions: Please read carefully**

* Please rename this file as only your ID number **(e.g. 20-\*\*\*\*\*-3.doc or 20-\*\*\*\*\*-3.pdf).**
* Submit the file before **11:59 PM on 13/12/2020** in VUES section labeled **Lab task-9. If you cannot complete the full task, do not worry. Just upload what you have completed.**

|  |
| --- |
| **Question No.1**  Create a class called person that contains protected members named name and age. Another class called employee extends or inherits the person class and contain protected members called joiningDate and salary. Another class called department inherits the employee class and contains protected members named id, deptName and position. Use the information to create employees for at least three different departments. |
| **Your code here:**  #include <iostream>  using namespace std;  int toa=0;  class person  {  protected:  string name;  int age;  public:  void setName (string a) {name=a;}  string getName () {return name;}  void setAge (int a) {age=a;}  int getAge () {return age;}  };  class employee: public person  {  protected:  string joiningDate;  double salary;  public:  void setJD (string a) {joiningDate=a;}  string getJD () {return joiningDate;}  void setSal (double a) {salary=a;}  double getSal() {return salary;}  };  class department : public employee  {  protected:  int id;  string deptName ;  string position;  public:  void setId (int a) {id=a;}  int getId () {return id;}  void setDN (string a) {deptName=a;}  string getDN () {return deptName;}  void setPos (string a) {position=a;}  string getpos () {return position;}  void print()  {  cout<<"Department: "<<deptName<<endl;  cout<<"Position: "<<position<<endl;  cout<<"Name: "<<name<<endl;  cout<<"ID: "<<id<<endl;  cout<<"Age: "<<age<<endl;  cout<<"Salary: "<<salary<<endl<<endl;  }  } ;  int main ()  {  department ob,ob1,ob2;  ob.setName ("Akash");  ob.setAge (21);  ob.setId (101);  ob.setDN ("Science and Techmnoly");  ob.setPos ("Dept Head");  ob.setSal (123456);  ob.print();  ob1.setName ("Avro");  ob1.setAge (25);  ob1.setId (102);  ob1.setDN ("Science and Techmnoly");  ob1.setPos ("Manger");  ob1.setSal (23456);  ob1.print();  ob2.setName ("Alen");  ob2.setAge (38);  ob2.setId (103);  ob2.setDN ("Science and Techmnoly");  ob2.setPos ("Clerk");  ob2.setSal (3456);  ob2.print();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| **Question No.2**  Create different classes and show the concept of multiple inheritance |
| **Your code here:**  #include<iostream>  using namespace std;  class one  {  public:  one()  {  cout << "one (constructor)" << endl;  }  };  class two  {  public:  two()  {  cout << "two (constructor)" << endl;  }  };  class three: public two, public one  {  public:  three()  {  cout << "three (constructor)" << endl;  }  };  int main()  {  three();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question No.3**  Implement the classes as shown by the diagram below   |  | | --- | | Class Name - person | | #name  #id  #age  +person()  +setter and getter  +print() |  |  | | --- | | Class Name - Employee | | -salary  +employee  +setter and getter  +printEmpInfo() |     For the given classes the constructor of person class will assign id as an employee object is created. The best way to do this is to pass a value to the person constructor as soon as an employee object is created. Implement this technique. |
| **Your code here:**  #include <iostream>  using namespace std;  class person  {  private:  string name;  int id;  int age;  public:  void setName(string a){name=a;}  string getName(){return name;}  void setAge(int a){age=a;}  int getAge(){return age;}  person(int a){id=a;}  int getid(){return id;}  void print() {  cout <<"Name: " <<getName() <<endl<<"ID: "<<getid() << endl<<"Age: "<<getAge()<<endl;  }};  class employee:public person  {  protected:  int salary;  public:  employee(int a):person(a){}  void setSalary(int a){salary=a;}  int getSalary(){return salary;}  void employeeinfo()  {  cout<< "Salary: "<< salary<< endl;  }  };  int main ()  {  employee ob(101);  ob.setName("Akash");  ob.setAge(21);  ob.setSalary(25000);  ob.print();  ob.employeeinfo();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| **Question No.4**  Extend the implemented solution in question 3, so that the id is auto-generated by the system.  Note – You can create auto-number generation as per your logic or look for codes for ‘random number generation using rand() in C++’. |
| **Your code here:**  #include <iostream>  using namespace std;  class person  {  private:  string name;  int id;  int age;  public:  void setName(string a)  {  name=a;  }  string getName()  {  return name;  }  void setAge(int a)  {  age=a;  }  int getAge()  {  return age;  }  person(int a)  {  for(int i=0; i<10; i++)  {  int a=rand()%100;  id=a;  }  }  int getId()  {  return id;  }  int getid()  {  return id;  }  void print()  {  cout <<"Name: " <<getName() <<endl<<"ID: "<<getid() << endl<<"Age: "<<getAge()<<endl;  }  };  class employee:public person  {  protected:  int salary;  public:  employee(int a):person(a) {}  void setSalary(int a)  {  salary=a;  }  int getSalary()  {  return salary;  }  void employeeinfo()  {  cout<< "Salary: "<< salary<< endl;  }  };  int main ()  {  employee ob(101);  ob.setName("Akash");  ob.setAge(21);  ob.setSalary(25000);  ob.print();  ob.employeeinfo();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| **Question No.5**  Create individual classes to show the concept of compile time polymorphism |
| **Your code here:**  #include<iostream>  using namespace std;  class person  {  public:  void print()  {  cout<<"No parameter"<<endl;  }  void print(float a)  {  cout<<"Float: "<<a<<endl;  }  void print(string a)  {  cout<<"String: "<<a<<endl;  }  };  int main()  {  person ob;  ob.print();  ob.print(7.78);  ob.print("Akash");  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| **Question No.6**  Create individual classes to show the concept of runtime time polymorphism |
| **Your code here:**  #include <iostream>  using namespace std;  class one  {  public:  void print()  {  cout<<"One class"<<endl;  }  };  class second:public one  {  public:  void print()  {  cout<<"Second class."<<endl;  }  };  int main()  {  cout<<"Runtime Polymorphism: (Function overloading)"<<endl<<endl;  one ob;  ob.print();  second ob1;  ob1.print();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Question No.7**  Consider the classes given below   |  | | --- | | Class: person | | #name  #id  #email  #specialization  +print() |  |  | | --- | | Class: doctor | | #noSurgery  +print() |  |  | | --- | | Class: teacher | | #schoolName  +print() |     Implement the classes and create object for doctor and teacher classes to print information using the print() function. |
| **Your code here:**  #include<iostream>  using namespace std;  class person  {  protected:  string name;  string id;  string email;  string specialization;  public:  void setName(string a){name=a;}  string getName(){return name;}  void setId(string a){id=a;}  string getId(){return id;}  void setEmail(string a){email=a;}  string getEmail(){return email;}  void setSpec(string a){specialization=a;}  string getSpec(){return specialization;}  void print()  {  cout<<"Name: "<<name<<endl;  cout<<"ID: "<<id<<endl;  cout<<"E-mail: "<<email<<endl;  cout<<"Specialization: "<<specialization<<endl;  }  };  class Doctor:public person  {  protected:  double noSurgery;  public:  void setNosurgery(double a){noSurgery=a;}  double getNosurgery(){return noSurgery;}  void print()  {  cout<<"Name: "<<name<<endl;  cout<<"ID: "<<id<<endl;  cout<<"E-mail: "<<email<<endl;  cout<<"Specialization: "<<specialization<<endl;  cout<<"Numeber of Surgery: "<<noSurgery<<endl;  }  };  class Teacher:public person  {  protected:  string schoolName;  public:  void setSchoolname(string a){schoolName=a;}  string getSchoolname(){return schoolName;}  void print()  {  cout<<"Name: "<<name<<endl;  cout<<"ID: "<<id<<endl;  cout<<"E-mail: "<<email<<endl;  cout<<"Specialization: "<<specialization<<endl;  cout<<"School Name: "<<schoolName<<endl;  }  };  int main()  {  cout<<"Doctor's info: "<<endl;  Doctor ob;  ob.setName("Asmaul Akash");  ob.setId("1003");  ob.setEmail("xxxxx@gmail.com");  ob.setSpec("Medicine");  ob.setNosurgery(42);  ob.print();  cout<<endl;  cout<<"Teacher's info: "<<endl;  Teacher ob1;  ob1.setName("Avro Alen");  ob1.setId("1005");  ob1.setEmail("yyyyy@gmail.com");  ob1.setSpec("Computer Science");  ob1.setSchoolname("Dhanmondi Gov't boys' School");  ob1.print();  } |
| **Your whole Screenshot here: (Console Output):** |