**Lab Task-8**

**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 29/11/2020** in VUES section labeled **Lab task 7. If you cannot complete the full task, do not worry. Just upload what you have completed.**

|  |
| --- |
| **Question No.1**  Create a class called test. The class will contain a private variable x. Add appropriate public members and a constructor to the class. |
| **Your code here:**  #include <iostream>  using namespace std;  class test  {  private:  int x;  public:  test(){cout<<"Constructor Added "<<endl;}  void setX (int a) {x=a;}  int getX () {return x;}  void print () {cout<<"Value of Variable: "<<x<<endl<<endl;}  };  int main()  {  test ob;  ob.setX(12345);  ob.print();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| **Question No.2**  Create a class to show the concept of constructor overloading. |
| **Your code here:**  #include <iostream>  using namespace std;  class ABC  {  private:  int x,y;  public:  ABC ()  { x = y = 0; }  ABC(int a)  { x = y = a;}  ABC(int a,int b)  { x = a; y = b; }  void display()  { cout << "x = " << x << " and " << "y = " << y << endl; }  };  int main()  {  ABC cc1;  ABC cc2(10);  ABC cc3(10,20);  cc1.display();  cc2.display();  cc3.display();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question No.3**  Implement the classes as shown by the diagram below   |  | | --- | | Class Name - Person | | -name  -age  -dob  +print() |  |  | | --- | | Class Name - Employee | | -salary  +printEmpInfo() | |
| **Your code here:**  #include <iostream>  using namespace std;  class person  {  private:  string name;  int age;  string dob;  public:  void setName(string a){name=a;}  string getName(){return name;}  void setAge(int a){age=a;}  int getAge(){return age;}  void setDob(string a){dob=a;}  string getDob(){return dob;}  void print() {  cout << "Name: " << getName() <<endl << "Age: " << getAge() << endl<< "Date of Birth: " << getDob()<<endl;  }};  class employee:public person  {  private:  float sal;  public:  void setSal(float a){sal=a;}  float getSal(){return sal;}  void printEmpInfo()  {  cout << "Salary: " << getSal () <<endl;  }  };  int main ()  {  employee ob;  ob.setName("John");  ob.setAge(29);  ob.setDob("18-November-1999");  ob.setSal(12343.76);  ob.print();  ob.printEmpInfo();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question No.4**  Implement the classes as shown by the diagram below. Later calculate the area for rectangle. **Make sure** that there will be no **garbage** values.   |  | | --- | | Class Name - Shape | | -length  -width  -height  +print() |   .   |  | | --- | | Class Name - Rectangle | | -area  *(length X Width)*  +printRecArea() | |
| **Your code here:**  #include <iostream>  using namespace std;  class shape  {  protected:  float length;  float width;  float height;  public:  void setL(float a){length=a;}  float getL(){return length;}  void setW(int a){width=a;}  float getW(){return width;}  void setH(float a){height=a;}  float getH(){return height;}  void print()  { cout << "Length: " << getL() <<endl << "Width: " << getW() << endl<< "Height: " << getH()<<endl;}  };  class rectangle:public shape  {  private:  float area;  public:  rectangle(){height=0;}  void setA( float area );  float getA()  { float a; float b;  a=length;  b=width;  area=length\*width;  return area;}  void printRecArea() {cout<<"Area: "<<getA()<<endl;}  };  int main()  {  rectangle ob;  ob.setL (25);  ob.setW(10);  ob.print();  ob.printRecArea();  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| **Question No.5**  Extend your solution in no.4 by adding a **Triangle Class** and a **Square Class. Make sure** that there will be no **garbage** values. Use the following formula-  *Area of triangle – ½ X length X Width*  *Area of Square – length X length* |
| **Your code here:**  #include <iostream>  using namespace std;  class shape  {  protected:  float length;  float width;  float height;  public:  void setL(float a){length=a;}  float getL(){return length;}  void setW(int a){width=a;}  float getW(){return width;}  void setH(float a){height=a;}  float getH(){return height;}  void print()  { cout << "Length: " << getL() <<endl << "Width: " << getW() << endl<< "Height: " << getH()<<endl;}  };  class rectangle:public shape  {  private:  float rarea;  public:  rectangle(){height=0;}  void setA( float rarea );  float getA()  { float a; float b;  a=length;  b=width;  rarea=length\*width;  return rarea;}  void printRecArea() {cout<<"Area of Rectangle: "<<getA()<<endl;}  };  class triangle:public shape  {  private:  float tarea;  public:  triangle(){height=0;}  void setT( float tarea );  float getT()  { float a; float b;  a=length;  b=width;  tarea= 0.5\*length\*width;  return tarea;}  void printTriArea() {cout<<"Area of Triangle: "<<getT()<<endl;}  };  class square:public shape  {  private:  float sarea;  public:  square() {height=0;}  void setS( float sarea );  float getS()  { float a; float b;  a=length;  b=width;  sarea= length\*length;  return sarea;}  void printSqArea() {cout<<"Area of Square: "<<getS()<<endl<<endl;}  };  int main()  { rectangle ob;  ob.setL (25);  ob.setW(10);  ob.print();  ob.printRecArea ();  cout<<endl;  triangle ob1;  ob1.setL (25);  ob1.setW(10);  ob1.print();  ob1.printTriArea();  cout<<endl;  square ob2;  ob2.setL (25);  ob2.print();  ob2.printSqArea ();  } |
| **Your whole Screenshot here: (Console Output):** |