**PSG College of Technology**

**Department of Applied Mathematics and Computational Sciences**

**Object oriented Programming with C++ Laboratory**

**Problem sheet**

**INHERITANCE**

1. Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data.

Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

2. Create a base class named Book. Data fields include title and author; functions include those that can set and display the fields. Derive two classes from the Book class: Fiction, which also contains a numeric grade reading level, and NonFiction, which contains a variable to hold the number of pages. The functions that set and display data field values for the subclasses should call the appropriate parent class functions to set and display the common fields, and include specific code pertaining to the new subclass fields.

Write a main()function that demonstrates the use of the classes and their functions. Save the file as **Books.cpp**.

3. Create a class named MusicalComposition that contains fields for title, composer, and year written. Include a constructor that requires all three values and an appropriate display function. The child class NationalAnthem contains an additional field that holds the name of the anthem’s nation. The child class constructor requires a value for this additional field. The child class also contains a display function. Write a main()function that instantiates objects of each class and demonstrates that the functions work correctly. Save the file as **Compositions.cpp**.

4**.** Complete the following tasks:

a. Create a base class named Rectangle that includes data members for the length and width of a Rectangle, as well as functions to assign and display those values. Derive a class named Block that contains an additional data member to store height, and contains functions to assign and display the height. Write a main()function that demonstrates the classes by instantiating and displaying the values for both a Rectangle and a Block. Save the file as **RectangleAndBlock.cpp**.

b. Add a member function to the Rectangle class that computes the area of a Rectangle (length multiplied by width). Add a member function to Block that has the same name, but overrides the computation with a volume calculation (length by width by height). Write a main()function that demonstrates the classes. Save the file as **RectangleAndBlock2.cpp**.

5. Create an Investment class that contains fields to hold the initial value of an investment, the current value, the profit (calculated as the difference between current value and initial value), and the percent profit (the profit divided by the initial value). Include a constructor that requires initial and current values and a display function. Create a House class that includes fields for street address and square feet, a constructor that requires values for both fields, and a display function. Create a HouseThatIsAnInvestment class that inherits from Investment and House. It includes a constructor and a display function that calls the display functions of the parents.

Write a main()function that declares a HouseThatIsAnInvestment and displays its values. Save the file as **HouseThatIsAnInvestment.cpp**.

6. **Ship , CruiseShip , and CargoShip Classes**

Design a Ship class that has the following members:

• A member variable for the name of the ship (a string)

• A member variable for the year that the ship was built (a string)

• A constructor and appropriate accessors and mutators

• A virtual print function that displays the ship’s name and the year it was built.

Design a CruiseShip class that is derived from the Ship class. The CruiseShip class should have the following members:

• A member variable for the maximum number of passengers (an int )

• A constructor and appropriate accessors and mutators

• A print function that overrides the print function in the base class.

The CruiseShip class’s print function should display only the ship’s name and the maximum number of passengers.

Design a CargoShip class that is derived from the Ship class. The CargoShip class

should have the following members:

• A member variable for the cargo capacity in tonnage (an int ).

• A constructor and appropriate accessors and mutators.

• A print function that overrides the print function in the base class. The CargoShip

class’s print function should display only the ship’s name and the ship’s cargo capacity.

7. Create a RestaurantMeal class that holds the name and price of a food item served by a restaurant. Its constructor requires arguments for each field. Create a HotelService class that holds the name of the service, the service fee, and the room number to which the service was supplied. Its constructor also requires arguments for each field. Create a RoomServiceMeal class that inherits from both RestaurantMeal and HotelService. Whenever you create a RoomServiceMeal object, the constructor assigns the string“room service” to the name of the service field, and $4.00 is assigned to the service fee inherited from HotelService. Include a RoomServiceMeal function that displays all of the fields in a RoomServiceMeal by calling display functions from the two parent classes. Additionally, the display function should display the total of the meals plus the room service fee. In a main()function, instantiate a RoomServiceMeal object that inherits from both classes. For example, a “steak dinner” costing $19.99 is a “room service” provided to room 1202 for a $4.00 fee. Save the file as **RoomService.cpp**.

8. Create a Painting class that holds the painting title, artist name, and value. All Paintings are valued at $400 unless they are FamousPaintings. Include a display function that displays all fields. The FamousPainting subclass overrides the Painting value and sets each Painting’s value to $25,000. Write a main()function that declares an array of 10 Painting objects. Prompt the user to enter the title and artist for each of the 10 Paintings. Consider the Painting to be a FamousPainting if the artist is one of the following: Degas, Monet, Picasso, or Rembrandt. Display the 10 Paintings. Save the file as **Paintings.cpp**.

9. Design a class Computer which contains memory size, processor name and network address as data members. Create another class Laptop which is a type of computer with the above properties and include weight as an additional data member, Create one more class Palmtop which contain the properties of both computer and laptop, and includes width and height as its data member. Write all suitable constructors for all classes. Write a friend function to check whether the network address is assigned or not. That friend function should be applicable to all the classes.

10. Write a base class worker and derived classes Hourlyworker and Salariedworker. Every worker has a name and a salary rate. Write virtual function compute\_pay(int hours) that computes the weekly pay for every worker. An hourly worker gets paid the hourly wage for the actual number of hours worked, if an hour is at most 40, if the hourly worker worked more than 40 hours, the excess is paid at time and a half. The salaried worker gets paid the hourly wage 40 hours no matter what the actual number of hours is.