Object Oriented Programming

Topic:

Polymorphism,
Composition, Function
Template

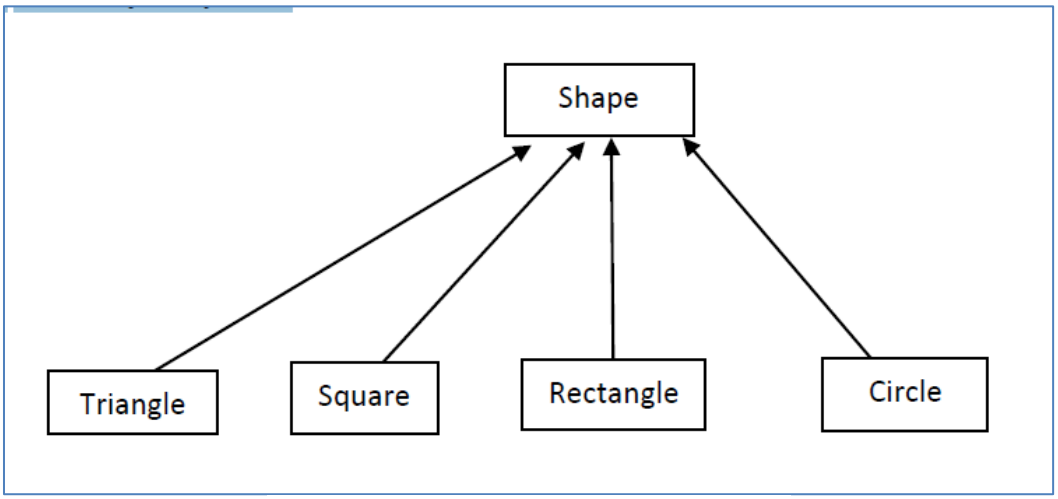
Faculty of Information Technology

UCP Lahore Pakistan

Task 1

Implement the following class hierarchy. Write a function Area () to calculate area of each object of any Shape.

Use Polymorphism



For Triangle: A = ½*length*width

For Square: A = length*length

For Rectangle: A = length*width

For Circle: $A = \pi r^2$

- Select attribute of each class as required above.
- Make Abstract class where needed.
- Make an parameterized constructor that takes parameters required in each class
- Call Area() function of each class through concept of polymorphism.

Task 2 - Function Templating

For this part of the lab make a template out of the myMax function and test it on different data types.

- Start with the repl code provided to you.
- Compile and run the program to see how it works.
- Make a template out of myMax. Don't forget the return type.
- Modify the prototype appropriately.
- Test your myMax template on int, double, and string types.

When you are done your output should resemble this:

```
The max of 3 and 5 is 5
The max of 5.6 and 7.3 is 7.3
The max of donkey and apple is donkey
```

Task 3 - Composition

Make a Computer System class with objects of the following classes

1. monitor
2. CPU
3. Keyboard

The monitor must have following member variables

1. char type array CompanyName
2. int Size
3. float price

CPU must have following member variables

1. char type array CompanyName
2. int Speed
3. float price

Key Board must have following member variables

1. Char type array CompanyName
2. Int NumOfKeys
3. Float price

In main user should see any information about his/her system.

NOTE: Also write the constructor & destructor sequence for every class.

Task 4:

Perform the following tasks.

1. An abstract class called Creature
2. Classes Player and Monster (derived from Creature)
3. Classes WildPig and Dragon (derived from Monster)

In the Creature class

– Define a char* member, CreatureName, to store the class Creature's name.

– Two functions

void DoAction() : Print the action of the object, and the actions have to be different from different classes.

void DrawOnScreen() : Print the object's name and call DoAction() belonging to the same class.

The class definition of Creature is:

```
class Creature {
public:
Creature(string);
virtual void DoAction()const=0;
virtual void DrawOnScreen()const=0;
protected:
string CreatureName;
};
```

Implement the class Player, Monster, Dragon and Wildpig so that when execution the following code, the counsel shows the execution result as the following:

Console Output:

```
Player (Kick_Ass) is attacking!!
Monster (UFO) is doing monster stuff!!
WildPig (I'm_Hungry) is Running!!
Dragon (I'm_the_Boss) is breathing Fire!!!
Press any key to continue . . .
```

Main function:

```
int main(){
Player hero("Kick_Ass");
Monster mon("UFO");
WildPig pig("I'm_Hungry");
Dragon drag("I'm_the_Boss");
Creature* object[4];
object[0]=&hero;
object[1]=&mon;
object[2]=&pig;
object[3]=&drag;
object[0]->DrawOnScreen();
object[1]->DrawOnScreen();
object[2]->DrawOnScreen();
object[3]->DrawOnScreen();
return 0;}
```