

# Assignment 4

## Object Oriented Programing

Topic: Polymorphism

Total Marks: 100

Due Date: 14<sup>th</sup> January 2018 11:59:59pm

### Instructions

1. Attempt all questions.
2. Copy/Cheating is **STRICTLY PROHIBITED**. If anyone caught doing cheating simply whole assignment will be canceled.
3. You can discuss assignment questions with other students but write your OWN code. Don't share your code with anyone.
4. Late submission will cause 50% deduction of marks.
5. Email Address for submission: bcsf16oop@gmail.com
6. Best of luck 😊

```
class Zakat
{
protected:
    int num; // contains number of animal due in zakat
    double money; // contains money that is due in zakat
public:
    Zakat(); // set num = money = 0.0
    Zakat(int); // set num;
    Zakat(double); //set money;
    virtual void calculateZakat() = 0;
    virtual void display() const = 0; // display zakat
    // write getter/setter
};

class SheepZakat : public Zakat
{
protected:
    int tNum; // contains total number of sheeps;
public:
    SheepZakat(); // set total number to zero and call parent class default
    constructor
    virtual void calculateZakat(); // This function calculates zakat on total number
    of sheeps.
    virtual void display() const;
    /*
        This function should display in following format
        Total number of sheeps owned : tNum
        Number of sheeps need to given in zakat : Num
    */
};

class CowZakat : public Zakat
```

```

{
protected:
    int tOneYear; // contains total number of one year cow;
    int tTwoYear; // contains total number of two year cow;

public:
    CowZakat(); // set total number to zero and call parent class default constructor
    virtual void calculateZakat(); // This function calculates zakat on total number
of cows.
    virtual void display() const;
    /*
    This function should display in following format
    Total number of cows owned : tNum
    Number of one year cows need to given in zakat : Num
    Number of two year cows need to given in zakat : Num
    */
};

class CamelZakat : public Zakat
{
protected:
    int tCNum; // contains total number of Camels;
    int tSNum; // contains total number of Sheeps;

public:
    CamelZakat(); // set total number to zero and call parent class default
constructor
    virtual void calculateZakat(); // This function calculates zakat on total number
of camels.
    virtual void display() const;
    /*
    This function should display in following format
    Total number of camels owned : tNum
    Number of camels need to given in zakat : Num
    Number of sheeps need to given in zakat : Num
    */
};

class MoneyZakat : public Zakat
{
protected:
    double goldVal, silverVal, tradeVal;
public:
    MoneyZakat(); // set every data member to zero and call default constructor of
parent class
    MoneyZakat(double, double, double); // set every data member respectively and call
parent class default constructor
    virtual void calculateZakat(); // calculate zakat on Money value;
    virtual void display() const;
    /*
    This function should display in following format
    Total Gold value : goldVal
    Total Silver value : silverVal
    Total Trade things value : tradeVal
    Sum of all three things: goldVal + tradeVal + silverVal
    Zakat on total money value : money
    */
    // write all getter/setter

```

```
};
```

```
void display(const Zakat *obj); // display the zakat of respective object reference by  
obj pointer
```

Write following menu based main to test the program

- Calculate Zakat on Money
  - Take input in goldValue, silver value and trading material value and calculate zakat and print them
- Calculate Zakat on Animals
  - Calculate Zakat on Sheeps
    - Take input total number of sheeps, calculate and print zakat
  - Calculate Zakat on Cows
    - Take input total number of cows, calculate and print zakat
  - Calculate Zakat on Camels
    - Take input total number of cows, calculate and print zakat
- Exit

### Zakat Calculation Formula

Type	Formula	Marks
Money	$(\text{GoldVal} + \text{SilverVal} + \text{tradeVal}) / 40$	10
Sheeps	<p>There is no zakat on sheeps less than 40 When sheeps = 40 then 1 sheep will be given in zakat When sheeps = 121 then 2 sheeps will be given When sheeps = 201 then 3 sheeps will be given When sheeps = 400 then 4 sheeps will be given From 400 onwards there is one sheep need to give on every 100 sheeps For example 500 = 5 sheeps 600 = 6 sheeps 700 = 7 sheeps</p> <p><b>NOTE: This is a pattern DON'T hardcode condition. Please observe pattern and make logic accordingly</b></p>	20
Cows	<p>There is no zakat cows less than 30. When cows = 30 then 1 cow of <b>one year</b> will be given in zakat</p>	30

	<p>When cows = 40 then 1 cow of <b>two year</b> will be given in zakat</p> <p>When cows = 60 then 2 cows of <b>one year</b> will be given in zakat</p> <p>When cows = 70 then 1 cow of <b>one year</b> will be given + 1 cow of <b>two year</b> will be given</p> <p>When cows = 120 then there are two options</p> <ol style="list-style-type: none"> <li>1. 4 cows of <b>one year</b> can be given</li> <li>2. 3 cows of <b>two year</b> can be given</li> </ol> <p><b>Note:</b> On every 30 cows you need to give <b>one year</b> cow and on every 40 cows you need to give <b>two year</b> cows. In 120 there are 3 sets of 40 and 4 sets of 30 hence either 4 cows of one year or 3 cows of two year can be given.</p> <p><b>NOTE: This is a pattern DON'T hardcode condition. Please observe pattern and make logic accordingly</b></p>	
Camels	<p>If camels are less than 5 there is no zakat on them</p> <p>If camels = 5 then 1 sheep will be given in zakat</p> <p>If camels = 10 then 2 sheeps</p> <p>If camels = 15 then 3 sheeps</p> <p>If camels = 20 then 4 sheeps</p> <p>If camels = 25 to 34 then 1 camel</p> <p>If camels = 35 to 44 then 2 camel</p> <p>If camels = 45 to 50 then 3 camel</p> <p>If camels = 55 then 3 camels + 1 sheep</p> <p>If camels = 60 then 3 camels + 2 sheeps</p> <p>If camels = 65 then 3 camels + 3 sheeps</p> <p>If camels = 70 then 3 camels + 4 sheeps</p> <p>If camels = 75 to 84 then 4 camels</p> <p>If camels = 85 to 94 then 5 camels</p> <p>If camels = 95 to 100 then 6 camels</p>	40

	<p>If camels = 105 then 6 camels + 1 sheep</p> <p>If camels = 110 then 6 camels + 2 sheep</p> <p>If camels = 115 then 6 camels + 3 sheep</p> <p>If camels = 120 then 6 camels + 4 sheep</p> <p>If camels = 125 to 134 then 7 camels</p> <p>If camels = 135 to 144 then 8 camels</p> <p>If camels = 145 to 150 then 9 camels</p> <p>And the pattern goes on</p> <p><b>NOTE: This is a pattern DON'T hardcode condition. Please observe pattern and make logic accordingly</b></p>	
--	--	--