**Table of content**

**Pages**

**1 Explanation of Polymorphism , inheritance and Encapsulation.......................................................1**

**2 Flowchart............................................................................................................................................2**

**3 A class .................................................................................................................................................2**

**4 the Class house ..................................................................................................................................3**

**5 pseudoscode........................................................................................................................................5**

**6 A program that declare two house object...........................................................................................6**

**Background of object oriented programming**

These days new software is usually object-oriented that is the software is written using an abstraction called an object this is naturally much more to commercial software development then simply writing lines of code there is investigation of the business requirement analysis of the problem , design of the solution and so on the development because they reduce the amount of information that has to be understood and improve the communication between member of the development team

**A brief history of programming**

* Machine code programming using a binary numner
* Assembly language programming using Alphanumeric symbol or mnemonics language is translated into a machine code by a program called an **Assembler**
* High level language programming using on forten are COBOL that have high level construct such as type function loops and branches high level language later generation of programming language Are translated into machine code using a program called a **compiler**

**Question one**

**Polymorphism :** is the ability or an entity to behave in different forms . Take a real word Example : the army ant’s . There are different sizes of an ant’s is the same ant colony with different sizes and queen is the target one this is polymorphic behaviour where the entities have a unique feature while they share all other common attributes and behaviour .

**Inheritance :** is the mechanism by which an object acquires some properties of another object it support the concept or the hierarchical classification

**Encapsulation :** describe the idea of building data and method that work on data within one unit a technique for minimising interdependencies among separately written modules by defining strict external interface the external interface of a module servers as a contract between the module and it’s client and thus between the designer of the module and other designer .

1.2 A pseudoscode to represent the logic of a program .

2

**Flowchart**

Input hours pay rate

Input hours pay work

If hours worked < 0

Print =Hours work can not be less then zero

Calculate gross pay

Gross pay = Hourly pay work x Hours work

Display= Gross pay

3

End

1.enter an hourly pay

2. Enter Hours worked

3.if hours worked < 0

Else print hours worked

Cannot be less than zero

4. Calculate gross pay

5. Display Gross pay

END

1.3

1. enter an hourly pay

2.enter hours worked

3.if hours worked < 0

Else print hours worked

Cannot be less than zero

4.calculate gross pay

Gross pay = Hourly pay x Hours worked

5.Display gross pay

6.if income + 9 x > Gross pay

Print income tax cannot be greater than gross pay

4 Else: net pay= Gross pay –

Income tax

7.END

**Question Two**

2.1 Class house {

Private

String street Address ;

Float prices ;

Int bedroom ;

Int bathroom ;

Public

Void set street address (string new address )

{

Street address = new address

Void get street address ( ) : {

Return street address

}

Void set price ( Float x ) : {

Price = x

Void get price ( ) :

Return asking price

}

Void set bedroom ( int bedroom , int bathroom )

Rooms = bedrooms , bathrooms

Void get bedroom ( ) :

5

Return bedroom

Return bathroom

}

|  |
| --- |
| House |
| -steet address : string  -price : Float  -bedroom : int  -bathroom : int |
| -Void set street address(string) ;  -void street address ( ) ;  -void set price (Float x) ; |

+ Void get price ( )

+ Void set Rooms ( int bedroom ; int bathroom )

+ Void set Rooms ( ) ;

Pseudoscode

1.declare Class

“ house”

2.Call set address ( )

3. Output function

“ set Value”

6

“Act value”

2.2 int main ( )

house House 1 ;

House 1 ; set street address ( “23 Cole Rd” )

: get Street address ( ) ;

Street address = 23 Cole Rd

House. House 2 ,

House 2. ; Set street address ( “14 smith Street” )

: get Street address ( )

Street address = 14 Smith Street

House. House 1

House. 1 set price ( 400.000 )

Set price ( )

Price = R 400 . 00

House. House 1

House. 1 set Rooms ( 4 bedrooms , 2 bathrooms )

Set Rooms ( )

Rooms = ( 4 bedrooms , 2bathrooms )

house house 2

House 2 set price ( 500.000 )

Set price ( )

Price. = R500.00

7

House. House 2

House. 2 set Rooms ( 5 bedrooms , 2 bathrooms )

Set Rooms ( )

Rooms = Bedrooms , 2 bathrooms

**References**

**www.programming-code.com**

**The current**

**Defining and creating classes diagram**