**School Of computing**

**Practical 7: Classes and Objects**

**ST0502 Fundamentals of Programming**

**What you will learn / do in this lab**

* Use classes to model objects
* Create objects from classes
* Write object-based programming

1. Determine the output and/or identify the error in each of the following code segments:

a)

|  |
| --- |
| var bankacct = { (type: "Savings"),(amtBalance: 500),  (bank: "OCBC"),(status: "active") };  console.log(type + " account & balance is $" + amtBalance); |
| Output/ Error: |

b)

|  |
| --- |
| var bankacct = { type: "Savings", amtBalance: 500,  bank: "OCBC", status: "active" };  console.log(bankacct.type + " account & balance is $"  + bankacct.amtBalance); |
| Output/ Error:  Saving account & balance is $500 |

c)

|  |
| --- |
| var bankacct = { type: "Savings", amtBalance: 500,  bank: "OCBC", status: "active" };  const interest = 0.10;  bankacct.amtBalance += interest \* bankacct.amtBalance;  console.log("New balance is $" + bankacct.amtBalance); |
| Output/ Error:  New balance is $550 |

|  |  |
| --- | --- |
| a) | Use the information below, fill in the blank to define and create as single object using an object literal.  Object variable :box  Property name : color  Property value : blue  var box = { color : ‘blue’ }; |
| b) | Declare and create another box object, value of color is red.  Var box2 ={ color: ‘red’} |
| c) | Write one console.log statement to print the color of the two boxes. Hardcoding is not acceptable ie console.log(“blue and red”);  Sample output : blue and red  Console.log(box.color + ‘ and ‘ + box2.color) |
| d) | Declare and create an object variable ***sch*** using the following details:   |  |  | | --- | --- | | Property name | Property value | | name | SOC | | type | poly | | telephone | 67721900 | | noITcourses | 3 | | postCode | 139651 |   var sch = {  name :‘SOC’  type : ‘poly’  telephone :‘67721900’  noITcourses :3  postcode :‘139651’  } |
| e) | Using the object created in Part (d), write a console.log statement to produce the following output:  **SOC** offers **3** full time **poly** diploma courses. Contact number is **67721900** and the postal code is **139651**.  The values in bold below are extracted from the object ie hardcoding is not allowed.  E.g of hardcoding  console.log(“SOC offers 3 full time poly diploma courses. Contact number is 67721900 and the postal code is 139651.”)  Output required:  Console.log(sch.name + ‘ offers ‘ + sch.noITcourses + ‘ full time ‘ + sch.type + ‘ diploma courses. Contact number is ‘ + sch.telephone + ‘ and the postal code is ‘ + sch.postCode) |

1. Given the following objects in a JavaSript codes:

|  |
| --- |
| var club1 = { name: "Sports", members: 1600, fees: 50 };  var club2 = { name: "Community Service", members: 1200,fees: 0 };  var club3 = { name: "Foodies", members: 800, fees: 100 }; |

Write JavaScript to produce the following output. Do not hardcode your codes.

|  |  |
| --- | --- |
| a) | **Output:** Fees to join Sports : $50  console.log(‘Fees to join ‘ + club1.name +’ : ‘ + club1.fees) |
| b) | **Output**: Total fees to join all the clubs are $150  Console.log( ‘Total fees to join all the clubs are $’ + (club1.fees + club2.fees + club3.fees)) |
| c) | Assume that you do not know the actual number of members in each club. Use a simple ***if*** statement to check whether Sports club has more or less members than Community Service club before writing the console.log statement.  **Output**: Sports is more popular than Community Service club  If(club1.members > club2.members){  Console.log(club1.name + ‘ is more popular than ‘ + club2.name + ‘ club.’)  } |
| d) | **Output:** Community Service has 400 more members than Foodies    Console.log(club2.name + ‘ has ‘ + (club2.members-club3.members) + ‘ more members than ‘ + club3.name +’ club.’) |

You may open up your Visual Studio Code (VSC) and work from there for the remaining

questions.

1. Use new Object () to create club4 with the following property values. Display the number of members in this new club.

|  |  |
| --- | --- |
| Property name | Property value |
| name | Arts |
| members | 700 |
| fees | 50 |

5. a) Create 2 objects , ***module*** and ***class1*** using the following properties. You may use any methods to create the objects.

|  |  |
| --- | --- |
| Object name : ***module*** | |
| Property name | Property value |
| code | ST0502 |
| abbr | FOP |
| year | 1 |
| tot\_class | 26 |
| tot\_students | 539 |

|  |  |
| --- | --- |
| Object name : ***class1*** | |
| Property name | Property value |
| class | Z01 |
| students | 20 |
| female | 12 |
| male | 8 |

b) Write codes to output the following :

i) Total number of classes taking FOP.

ii) Percentage of female students in class1 taking FOP.

iii) Only the top 5% of the total students are eligible for distinctions. Write a function ***computeDist*** within object ***module*** to calculate the number of students eligible for distinctions.

varmodule={

*code*:'ST0502',

*abbr*:'FOP',

*year*:'1',

*tot\_class*:26,

*tot\_students*:539,

*computeDist*(){

*return* (5/100\* *this.tot\_students*)

}

}

varclass1={

*class*:'Z01',

*students*:20,

*female*:12,

*male*:8

}

*console.log*('Total number of classes taking ' + module*.*abbr + ': ' + module*.*tot\_class)

*console.log*((*class1.*female/module*.*tot\_students)\*100 + '%')

*console.log*(module*.computeDist*())

1. Given the following JavaScript codes:

|  |
| --- |
| var sch = {      name: "SOC",      type: "poly",      tel: "67721900",      noITcourses: 3,      postCode: '139651',      printTel: function () {          return this.name + ":" + this.tel;      }  };  console.log("Telepone of " + sch.printTel()); |

|  |  |
| --- | --- |
| a) | What is the output? |
| b) | Wite another method to print the postal code for SOC. |

1. a) Using the Question 3, create a class with constructor to create the same objects. Without using a method in the class, write a statement to output the total fees to join all 3 clubs.

b) Without using a method, code 3 console.log statements to print the joining fee for each club. If foreigners need to pay 10% more to join any of the club, amend your codes and calculate and display the foreign fee for each club.

c) Rewrite part (b) using a method statement within the class to calculate the fees foreigners need to pay to join each club. Display the foreign fees of each club.

d) What is the difference between (b) and (c). Name one advantage/disadvantage.

Advantage of methods: don’t need to calculate in every statement

8. a) Create a ***Rectangle*** class to simulate a Rectangle. Within the class, you should:

1. Two properties, ***length*** and ***width***.
2. A method ***getArea()*** that returns the area of the Rectangle.
3. Another method ***getPerimeter()*** that returns the perimeter of the Rectangle.

b) In the main program,

1. Create an instance of the Rectangle class named ***r1***.
2. Set the ***length*** of ***r1*** to be 10.
3. Set the ***width*** of ***r1*** to be 5.
4. Display the ***area*** and ***perimeter*** of ***r1***.

Program output:

|  |
| --- |
| Area of rectangle r1 is 50  Perimeter of rectangle r1 is 30 |

class Rectangle {

    constructor(length,width){

        this*.*length = length

        this*.*width = width

    }

*getArea*(){

*return* this*.*length\*this*.*width

    }

*getPerimeter*(){

*return* (this*.*length+this*.*length+this*.*width+this*.*width)

    }

}

// *part b*

varr1=new *Rectangle*(10,5)

*console.log*('Area of rectangle r1 is ' + *r1.getArea*())

*console.log*('Perimeter of rectangle r1 is ' + *r1.getPerimeter*())

9. a) Create a ***Circle*** class to simulate a Circle. Within the class, you should have:

* 1. One property, ***radius***.
  2. A method ***getArea()*** which returns the circle's area. You should use the constant ***PI*** from the ***Math*** library.
  3. A method ***enlargeCircle()*** such that the circle’s radius will be tripled.
  4. A method  ***shrinkCircle()*** such that the circle’s radius will be halved.

b) In the main program,

1. Create an instance of Circle named ***circle1*** with ***radius*** of 2.
2. Display the area of ***circle1*** as shown in the program output.
3. Enlarge the radius of ***circle1*** by 3 times and display the area of ***circle1*** as shown in the program output.
4. Halve the radius of ***circle1*** and display the area of ***circle1*** as shown in the program output.

Program output:

|  |
| --- |
| Area of circle1 with radius 2.0 is 12.566370614359172  Circle is enlarged 3 times.  Area of circle1 with radius 6.0 is 113.09733552923255  Circle is shrunk by halve.  Area of circle1 with radius 3.0 is 28.274333882308138 |

// *Question 9*

// *part a*

class Circle {

    constructor(radius){

        this*.*radius = radius

    }

*getArea*(){

*return* *Math.*PI\*(this*.*radius\*\*2)

    }

*enlargeCircle*(){

*return* this*.*radius \*3

    }

*shrinkCircle*(){

*return* this*.*radius/2

    }

}

// *part b*

varcircle1=new *Circle*(2)

*console.log*('Area of circle1 with radius ' + *circle1.*radius + ' is ' + *circle1.getArea*())

*console.log*('Circle is enlarged 3 times')

*circle1.*radius=*circle1.enlargeCircle*(*circle1.*radius)

*console.log*('Area of circle1 with radius ' + *circle1.*radius + ' is ' + *circle1.getArea*())

*console.log*('Circle is shrunk by halve')

*circle1.*radius=*circle1.shrinkCircle*(*circle1.*radius)

*console.log*('Area of circle1 with radius ' + *circle1.*radius + ' is ' + *circle1.getArea*())

10. a) Write ***BankAccount*** class that will be used to create bank accounts for all users of the bank. The class consists of the following:

1. Two properties, ***name*** (String type) and ***savings*** (float type).
2. Write the method ***getBalance()*** that returns a String value in the following format:

*<name> + “has $”+ <savings>*

For example,

Tom has $1234.0

b) In the main program, write codes to generate the following output:

Program output:

|  |
| --- |
| Oliver Twist has $1000.00  Richie Rich has $100000.00  // *question 10*  // *part a*  class BankAccount{      constructor(name, savings){          this*.*name = *name.toString*()          this*.*savings = *savings.toFixed*(2)      }  *getBalance*(){  *return* this*.*name + ' has $' + this*.*savings      }  }  // *part b*  varperson1=new *BankAccount*('Oliver Twist',1000.00)  varperson2=new *BankAccount*('Richie Rich',100000.00)  *console.log*(*person1.getBalance*())  *console.log*(*person2.getBalance*()) |

(Optional )

11. Write a class named ***Fan*** to model fans. The properties of the ***Fan*** class are ***speed*** and ***on*** of ***integer*** type and ***boolean*** type respectively.

The values of ***speed***: ***1***, ***2*** and ***3*** denote the speed of the fan as slow, medium and fast respectively.

The value of ***on***: ***true*** denotes the fan is on and ***false*** denotes the fan is off.

Code a method ***getState()*** that returns the state of the fan object as shown.

|  |
| --- |
| For example if ***speed*** is ***2*** and ***on*** is ***true***, invoking ***showState()*** will return:  ***on*** at ***medium*** speed  For example if ***on*** is ***false***, invoking ***showState()*** will display:  ***off*** |

In the main program,

* + 1. Instantiate 2 Fan objects. The first fan has ***on*** status with ***low*** speed, and the second fan also has ***on*** status with ***fast*** speed.
    2. Invoke ***getState()*** method to display the state of both fans.
    3. Switch off the first fan and set the speed of the second fan to medium.
    4. Invoke ***getState()*** again to display the state of both fans.

Program output:

|  |
| --- |
| Fan 1 is on at low speed.  Fan 2 is on at fast speed.  Fan 1 is off.  Fan 2 is on at medium speed. |

**- END -**