Trent University
COIS1020H
Lab 5 Answer Document

1) Introduction to Objects (with Constructors and Properties)

b) Build and execute the program (F5) … resolve any syntax mistakes (if you make any) until the program compiles.

What is the output?

Employee 1: , 0, ,

Which constructor from the Employee class was invoked to instantiate `emp1` object?

Constructor 1 was invoked

```
public Employee() // Constructor 1: No Arg
{
        Name = "";
        IdNumber = 0;
        Department = "";
        Position = "";
}
```

Why are the contents of the `emp1` object appear to be empty when printed?

The constructor that was invoked to instantiate emp1 object had stored null in the strings Name, Department and position and thus these contents of emp1 were empty when printed.

c) Uncomment Lines 12-15 in the *EmployeeDemo* class to render them active

What is the output?

Employee 1: Susan Meyers, 10012, Accounting, Vice President

Why is the output for Employee 1 different that in Part (b)?

The lines that were uncommented called the properties in the Main and values were stored in them when initially they were given no value. So, when object emp1 is instantiated these values are

there and thus printed.

d) Uncomment Lines 8, 23, 24 in the *EmployeeDemo* class to render them active, execute the program.

What is the output?

Employee 1: Susan Meyers, 10012, Accounting, Vice President
Employee 2: Joy Rogers, 10127, HR, Manager

Which constructor from the Employee class was invoked to instantiate `emp2` object?

Constructor 3

```
 // Constructor 3: Four Arg
 public Employee(string eName, int eIdNum, string eDept, string ePos)
{
    Name = eName;
    IdNumber = eIdNum;
    Department = eDept;
    Position = ePos;
}
```

e) Uncomment Lines 9, 25 and 26 in the *EmployeeDemo* class to render them active, execute the program.

What is the output?

Employee 1: Susan Meyers, 10012, Accounting, Vice President
Employee 2: Joy Rogers, 10127, HR, Manager
Employee 3: Mark Jones, 10065, ,

Which constructor from the Employee class was invoked to instantiate `emp3` object?

Constructor 2

```
 public Employee(string eName, int eIdNum) // Constructor 2: Two Arg
{
       Name = eName;
       IdNumber = eIdNum;
       Department = "";
       Position = "";
}
```

f) Uncomment Lines 17 and 18 in the *EmployeeDemo* class to render them active, execute the program.

What is the output?

Employee 1: Susan Meyers, 10012, Accounting, Vice President
Employee 2: Joy Rogers, 10127, HR, Manager
Employee 3: Mark Jones, 10065, IT, Support

Why is the output for Employee 3 different that in Part (e)?

The lines that were uncommented called the properties in the Main and values were stored in them when initially they were given no value. So, when object emp3 is instantiated these values are there and thus printed.

g) Put a Breakpoint on Line 7, 8, and 9 in the *EmployeeDemo* class and execute the program.

What do you see in the *Locals* window?

Name	Value	Type
emp1	null	Employee
emp2	null	Employee
emp3	null	Employee

h) Use to make one step.

Where did you step to (be specific)?

The highlight was on the constructor 1header in the Employee class

```
public Employee() // Constructor 1: No Arg
 {
     Name = "";
     IdNumber = 0;
     Department = "";
     Position = "";
 }
```

i) Move to the next Breakpoint (should be on Line 8). Re-examine the Locals window.

How has it changed from Part (g)?

Name	Value	Type
◢ emp1	{Employee}	Employee
Department	""	string
IdNumber	0	int
Name	""	string
Position	""	string
department	""	string
idNumber	0	int
name	""	string
position	""	string
emp2	null	Employee
emp3	null	Employee

Now instead of null, the emp1 object has class Employee and the properties of the constructor that was called because complier went through that part.

j) Make one step.

Where did you step to (be specific)?

```
// Constructor 3: Four Arg
public Employee(string eName, int eIdNum, string eDept, string ePos)
{
    Name = eName;
    IdNumber = eIdNum;
    Department = eDept;
    Position = ePos;
}
```

The highlight reached the constructor 3 header in the Employee class.

k) Move to the next Breakpoint (should be on Line 9). Re-examine the Locals window.

How has it changed from Part (i)?

Name	Value	Type
◢ emp1	{Employee}	Employee
Department	""	string
IdNumber	0	int
Name	""	string

Position	""	string
department	""	string
idNumber	0	int
name	""	string
position	""	string
◢ emp2	{Employee}	Employee
Department	"HR"	string
IdNumber	10127	int
Name	"Joy Rogers"	string
Position	"Manager"	string
department	"HR"	string
idNumber	10127	int
name	"Joy Rogers"	string
position	"Manager"	string
emp3	null	Employee

The values in emp2 object has also changed because the complier went to Employee class and the properties in the constructor that was called.

l) Continue stepping through the program examining the contents of the *Locals* window and observing how the program transfers control between *Main* to *Employee*.

Comment on your experience.

As we press F11 after the highlight is on Line 9, it jumps to Employee class and to the constructor 2 which was called and goes through it. As it proceeds, the values in the local window changed for this reference. After this it jumps back in Main and to the line 9. Then it proceeds through the code in Main and the values keep changing in the local window for emp1 object properties and emp3 object properties. Since there are private properties, get and set have been used to change their value when we want to from Main, i.e., when we are through the code emp1.Name till emp3.Position (Line12-18 in Main)

m) Remove all thee breakpoints. In Line 13, change *IdNumber* to *idNumber* in the *EmployeeDemo* class.

What happens and why?

Error is shown.
That is because we are trying to access a private property from another class which is not possible without get and set. In the error list it shows inaccessible due to its protection level.

n) In Line 4 of the *Employee* class (note the different file here), change *private* to *public*.

What happens and why?

```
The program runs without any errors because idNumber is no longer
     private and hence no security problem for accessing it.
```

Is it a good idea to solve the problem this way? Why?

It is not a good idea to solve the program this way because then anyone would be able to access the idNumber of any person and there won't be any protection

Answer all the highlighted questions in a file and then submit a PDF of this file (called it Lab5_1.pdf) to the Lab 5 dropbox. When asked "What is the output", simply type in what is seen in the output window.

2) Introduction to Objects (with Constructors and Properties)

b) Run the program using input -2, 5

What is the syntax error and how did you fix it?

```
Error:
Console.WriteLine("Circle 1 has radius {0} and area {1:F2}", cir1.Radius, cir1.GetArea);
Correction:
Console.WriteLine("Circle 1 has radius {0} and area {1:F2}", cir1.Radius, cir1.GetArea());
```

What is the output?

Circle 1 has radius 0 and area 0.00
Circle 2 has radius 10 and area 314.16
Enter a positive radius => -2
Enter a positive radius => 5
Circle 1 has radius 5 and area 78.54
Circle 2 has radius 10 and area 314.16

c) Add the following lines of code after Line 30 in the *CircleDemo* class (Lab5_2_1.cs)

What is the output (assume an input of 5)?

```
Circle 1 has radius 0 and area 0.00
Circle 2 has radius 10 and area 314.16
Enter a positive radius => 5
Circle 1 has radius 5 and area 78.54
Circle 2 has radius 10 and area 314.16
Circle 1 has radius 0 and area 0.00
Circle 2 has radius 10 and area 314.16
```

How did the program arrive at this result for Circle 1?

The code we added said cir1.Radius=-8. So when we go in Circle class, and we have to change the value of the private property, we are using get and set. While in set we have if-else for validating the value being stored. So, the code says the value stored in cir1.Raius is -8, which is less than 0. And the if else condition says, if we have negative value then radius =0. So we arrive at that result for circle 1.

d) Add a new instance method to the *Circle* class in Lab5_2_2.cs to compute the circumference of a circle (recall that it is 2 * PI * radius).

Show the details of the *GetCircumference()* method?

```
public double GetCircumference()
 {
     double circumference;

     circumference = 2 * PI * radius;
     return circumference;
 }
```

Show the details of ONE of the modified *Console.WriteLine* statements?

```
Console.WriteLine("Circle 1 has radius {0} and area {1:F2} and circumference is {2:F2}",
cir1.Radius, cir1.GetArea(), cir1.GetCircumference());
```

What is the output when run with an input of 5?

Circle 1 has radius 0 and area 0.00 and circumference is 0.00
Circle 2 has radius 10 and area 314.16 and circumference is 62.83
Enter a positive radius => 5
Circle 1 has radius 5 and area 78.54 and circumference is 31.42
Circle 2 has radius 10 and area 314.16 and circumference is 62.83
Circle 1 has radius 0 and area 0.00 and circumference is 0.00
Circle 2 has radius 10 and area 314.16 and circumference is 62.83

e) Finally, let's look at including an overloaded + operator to our *Circle* class.

Show the details of the lines of code added to the *CircleDemo* class?

```
cir3 = cir1 + cir2;

Console.WriteLine("Circle 3 has radius {0} and area {1:F2} and circumference is {2:F2}",
cir3.Radius,
cir3.GetArea(), cir3.GetCircumference());
```

What is the output when run with an input of 5?

Circle 1 has radius 0 and area 0.00 and circumference is 0.00
Circle 2 has radius 10 and area 314.16 and circumference is 62.83
Enter a positive radius => 5
Circle 1 has radius 5 and area 78.54 and circumference is 31.42
Circle 2 has radius 10 and area 314.16 and circumference is 62.83
Circle 1 has radius 0 and area 0.00 and circumference is 0.00
Circle 2 has radius 10 and area 314.16 and circumference is 62.83
Circle 3 has radius 5.64189821823184 and area 100.00 and circumference is 35.45

Answer all the highlighted questions in a file and then submit a PDF of this file (called it Lab5_2.pdf) to the Lab 5 dropbox. When asked "What is the output", simply type in what is seen in the output window.

3) Putting it All Together

Demonstrate to the Lab personnel and Submit the .cs file to the Lab 5 dropbox.