

Examination Answer Book

12 Pages

Instructions to candidates

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right and seal to hide your name.
3. The answers must be **legibly** written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

1	Student Number	21419075	List the numbers of the questions you have attempted	Office Use
2	Title of Examination Paper	Apparatus Development	7	18
3	Module Code	CSAC5001	8	3
4	Date and Time of Examination	11-31-2019	9	4
5	Examination Venue	Coleg y Gŵr	10	0
6	Seat Number		(10/10)	
7	Number of Answer Books handed in	2	Book 1	40
8			Book 2	25
9			Total	65

1	Rhif Myfyrwr		Rheswch rhifur cwestiynau a ymgaiswyd gennych	At Ddibenion y Swyddfa
2	Teitl y Papur Arholiad			
3	Cod y Modwl			
4	Dyddiad ac Amser yr Arholiad			
5	Lleoliad yr Arholiad			
6	Rhif Sedd			
7	Nifer y Llyfrau Atebion a gyflwynwyd		Cyfanswm	



Llyfr Atebion Arholiadau

12 Tudalen

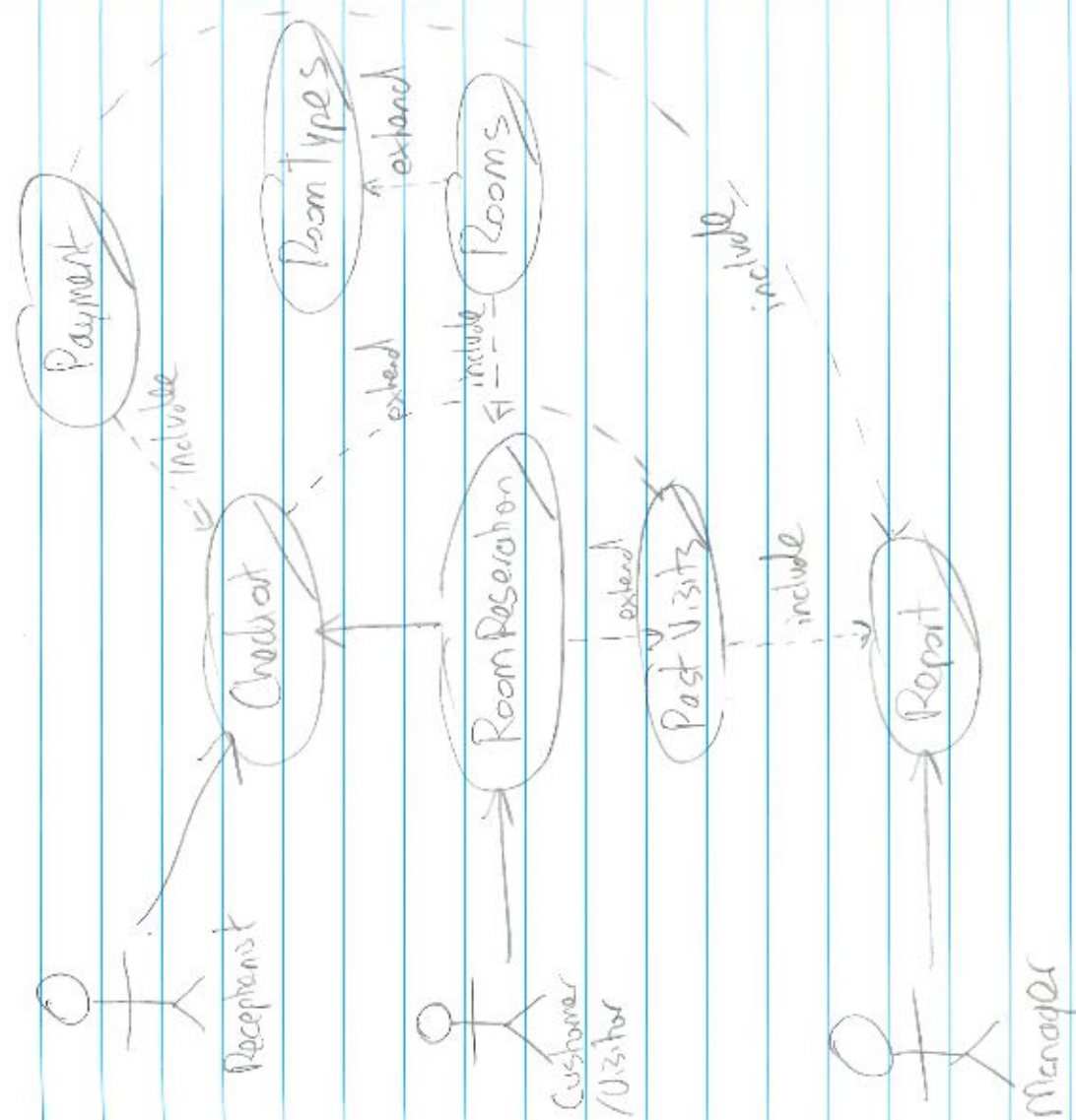
Cyfarwyddiadau i Ymgaiswyr

1. Rhwch yr holl wybodaeth y gofynnir andani yn y blychau isod.
2. Rhwch eich enw yn y lle a ddarperir i'r dde a seliwch i guddio eich enw.
3. Rhaid i'r atebion fod mewn llawysgrif **ddarlennadwy**. Gall yr Arholwyr naill ai anwybyddu sgrifl amharlennadwy, neu drefnu tiddi gael ei thcipio a chael yr ymgaisydd i dalu.
4. Ysgrifennwch eich atebion mewn inc.
5. Dechreuwch bob ateb ar dudalen newydd. Gwnewch unrhyw fras nodiadau yn y llyfr atebion a thynnu llinell drwyddynt.
6. Ni ddylid rhwygo unrhyw ran o'r llyfr hwn; rhaid ei gyflwyno yn gyfan.
7. Os defnyddir mwy nag un llyfr, rhwymwch hwy gyda'i gilydd gan ddefnyddio'r tagiau sydd ar gael oddi wrth y gwyllwr/gwyllwyr.
8. Rhaid dychwelyd pob llyfr. Ni chaniateir mynd â llyfrau atebion, wedi eu defnyddio neu heb eu defnyddio, allan o'r lleoliad arholi o dan unrhyw amgylchiadau.

A wnewch chi lenwi'r rhan hon a'i SELIO cyn i'r arholiad ddechrau.

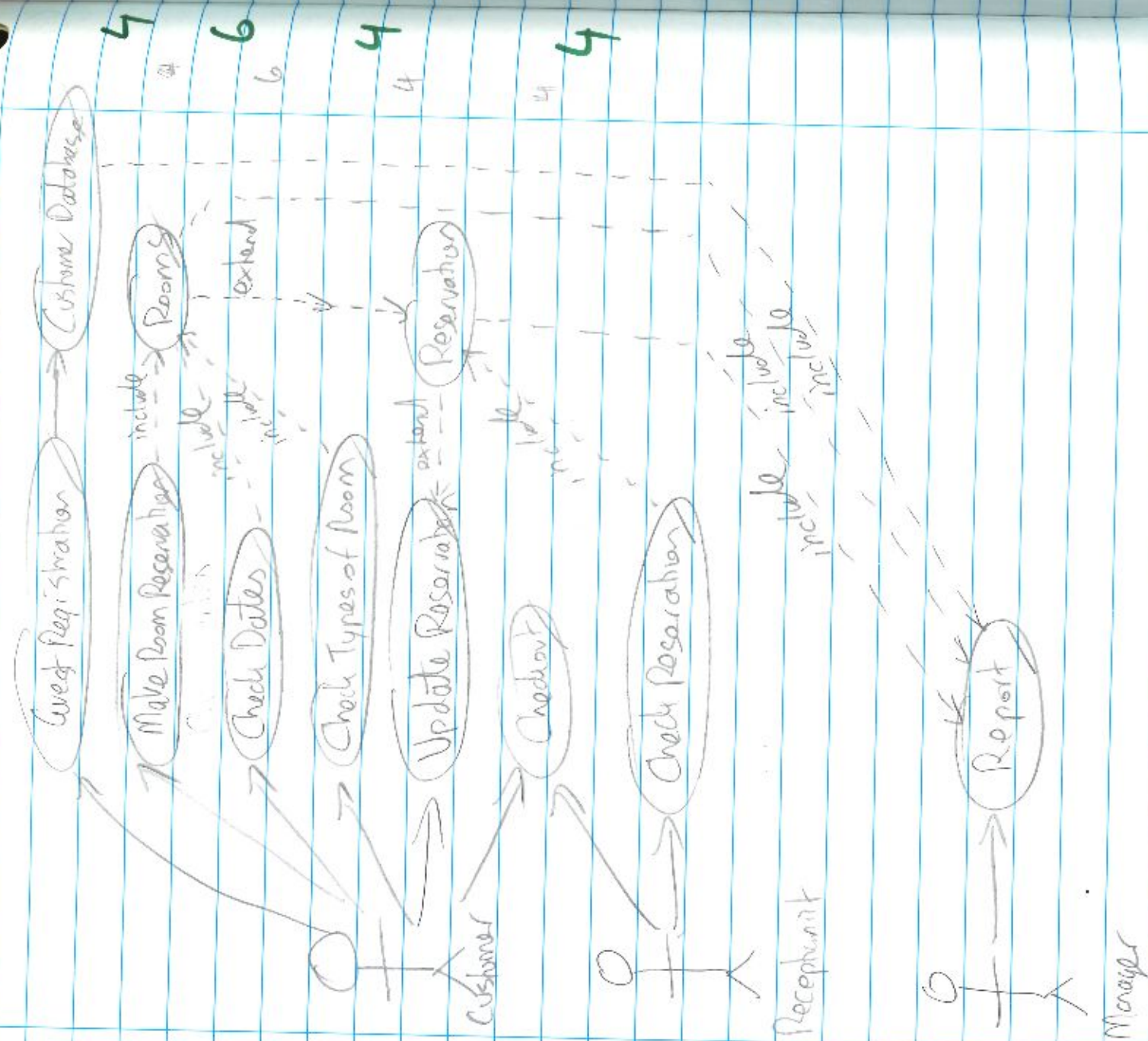
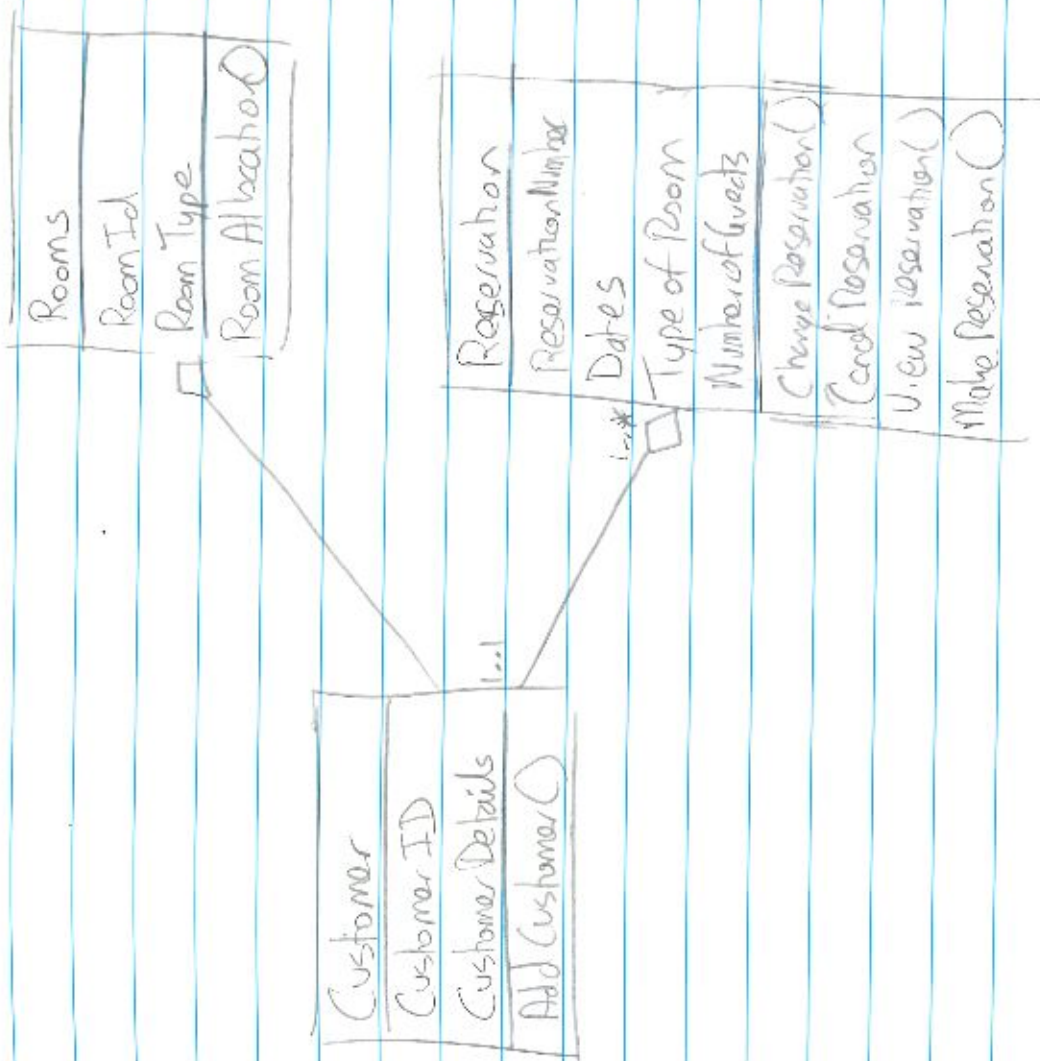
Cyfenw

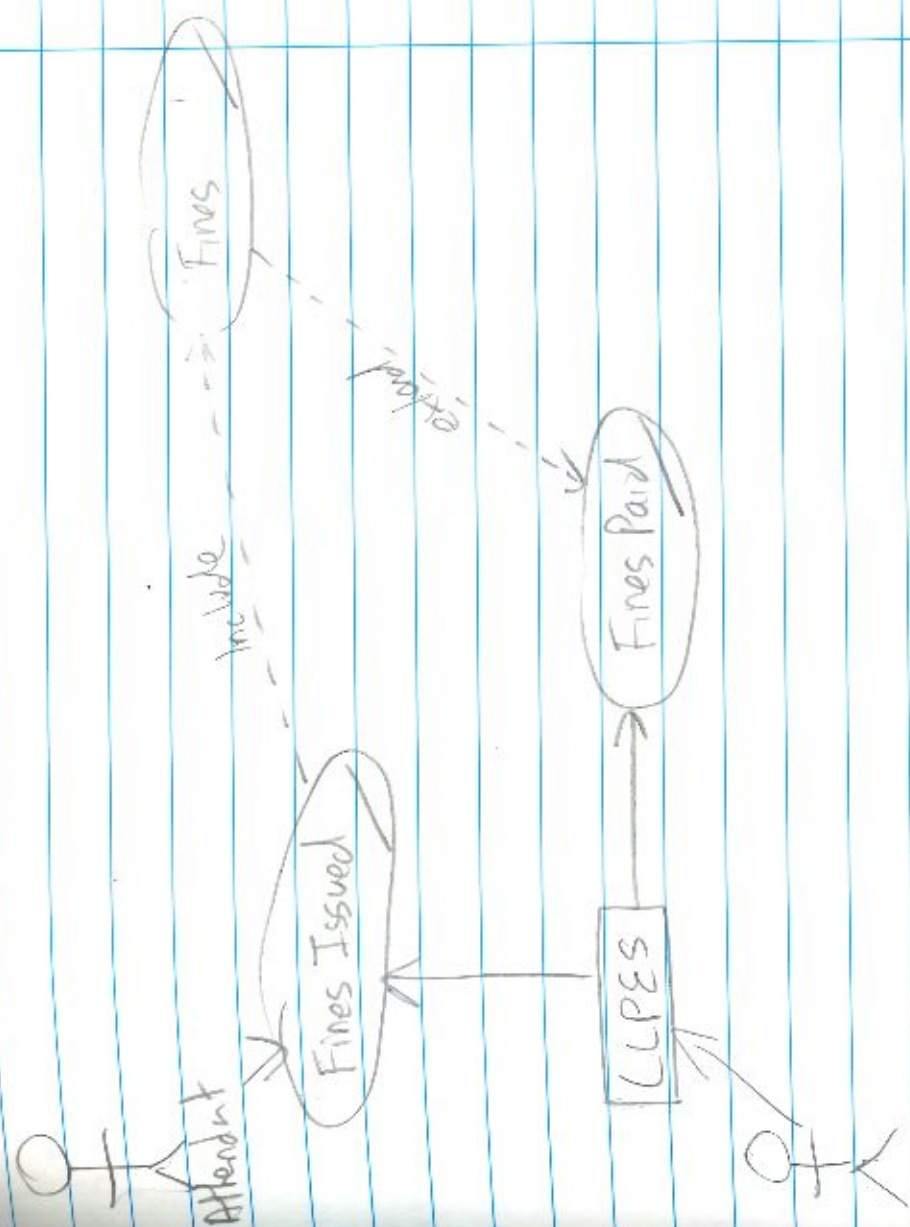
(LLYTHRENNAU BREISION) Enw(au) cyntaf



Page

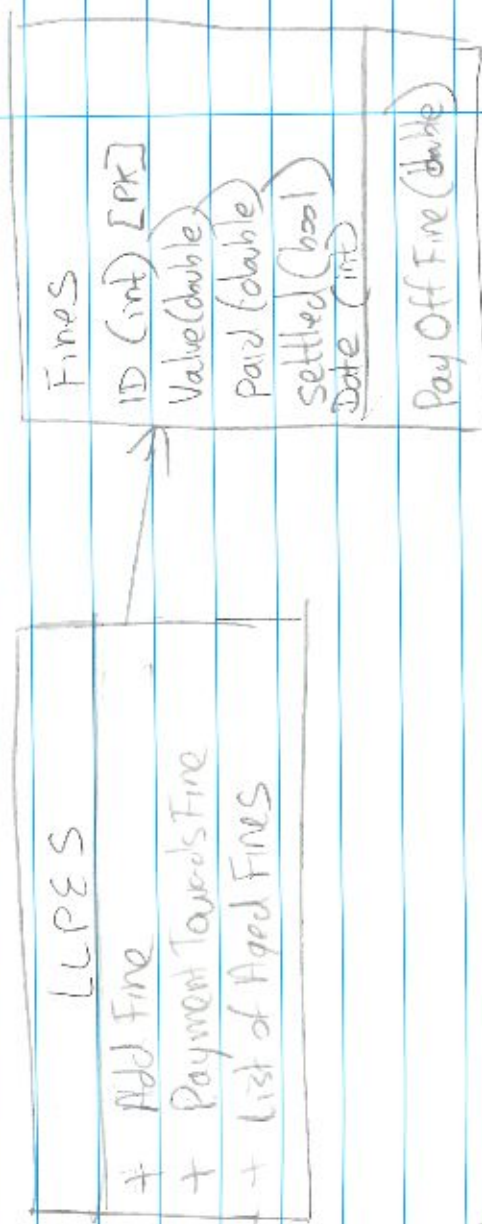
220





3

3



(i) $1..*$

Where there is at least 1 instance (no upper limit)
Lower Bound of 1, no upper bound

(ii) $0..*$

Can mean 0 or more instances (no upper limit)
Lower Bound of 0, no upper bound

(iii) $0..1$

There can be none or exactly one instance only
Lower Bound of 0, Upper Bound of 1.

(iv) 2

There can only be exactly 2 instances

(v)

There can be any number represented.

10 (i) + Sets A value

(ii) - Gets A value

(iii) - Private

(iv) / or

(v) # Number of

Surname

ROBERTS

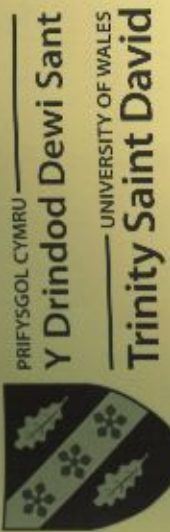
(Block Capitals) Forename(s)

Stephen

Please complete this section and SEAL before the examination starts.

Complete
section,
remove
strip, fold as
indicated
and SEAL

FOLD



Examination Answer Book

12 Pages

Instructions to candidates

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right and seal to hide your name.
3. The answers must be **legibly** written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

List the numbers of the questions you have attempted	Office Use
1	10
2	5
3	9+2+6
5	8
6	11
(other pages)	
7	
8	
9	
10	
Total	40

1	Student Number	92061412
2	Title of Examination Paper	Application Design & Development
3	Module Code	CSAC5001
4	Date and Time of Examination	9th January 18 11.30am
5	Examination Venue	Colleg yr Gwr
9	Seat Number	
7	Number of Answer Books handed in	2

(Gweler yr ochr arall am y fferiwn Cymraeg)

40

PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

Llyfr Atebion Arholiadau

12 Tudalen

Cyfarwyddiadau i Ymgeiswyr

1. Rhwch yr holl wybodaeth y gofynnir andani yn y blychau isod.
2. Rhwch eich enw yn y lle a ddarperir i'r dde a seliwch i guddio eich enw.
3. Rhaid i'r atebion fod mewn llawysgriflen **ddarlennadwy**. Gall yr Arholwyr naill ai anwybyddu sgrïpl annarlennadwy, neu drefnu iddi gael ei theipio a chael yr ymgeisydd i dalu.
4. Ysgrifennwch eich atebion mewn linc.
5. Dechreuwch bob ateb ar dudalen newydd. Gwnewch unrhyw fras nodiadau yn y llyfr atebion a thynnw llinell drwyddynt.
6. Ni ddylid rhwygo unrhyw ran o'r llyfr hwn; rhaid ei gyflwyno yn gyfan.
7. Os defnyddir mwy nag un llyfr, rhwymwch hwy gyda i gilydd gan ddefnyddio'r tagiau sydd ar gael oddi wrth y gwyliwr/gwyliwyr.
8. Rhaid dychwelyd pob llyfr. Ni chaniateir mynd â llyfrau atebion, wedi eu defnyddio neu heb eu defnyddio, allan o'r lleoliad arholi o dan unrhyw amgylchiadau.

1	Rhif Myfyrwr		Rhestrwch rifau'r cwestiynau a ymgeisiwyd gennych	At Ddibecion y Swyddfa
2	Teitl y Papur Arholiad			
3	Cod y Modwl			
4	Dyddiad ac Amser yr Arholiad			
5	Lleoliad yr Arholiad			
6	Rhif Sedd			
7	Nifer y Llyfrau Atebion a gyflwynwyd			

(See the other side for English version)

A wnewch chi lenwi'r rhan hon a'i SELIO cyn i'r arholiad ddechrau.

Cyfenw _____ (LLYTHRENAU BREISION) Enw(au) cyntaf _____

Llenwch y
rhan hon,
tynnwch y
sgrïpl,
plygwch fel
a ddangosir
a SELIOWCH

PLYGWCH

① (i) A class is a construct that allows you to create your own custom types by grouping together variables of other types, methods and events

An example =

```
public class MyClass
```

```
{
```

```
    // Fields, Properties, Methods, Events go here
```

```
}
```

2

2

(ii) An object is a block of memory that has been allocated and configured according to a class blueprint

An example

```
class Program
```

```
{
```

```
    static void Main()
```

```
{
```

```
        // Create object from MyClass
```

```
        MyClass newObject = new MyClass();
```

```
}
```

```
}
```

2

2

① continued

(iii) An abstract data type allows us to create classes and class members that are incomplete and must be implemented in a derived class. An abstract class cannot be instantiated, instead the purpose is to provide a common definition of a base class that many derived classes can share.

An example =

```
public class My Class
```

```
{
    public virtual void DoAction (var Variable)
```

```
{
    // Original code
```

```
}
```

```
public abstract class NewClass = myClass
```

```
{
    public abstract override void DoAction (var Variable);
```

```
}
```

```
public class absClass = NewClass
```

```
{
    public override void DoAction (var Variable);
```

```
{
    // New code to override original code
```

```
}
```

```
}
```

5

2

① continued

(iv)

~~Method overloading~~
Method overloading means creating two or more methods of the same type that differ only in the number or type of parameters but have the same name.

When two or more methods in a type are the same kind of member (method, property, constructor etc) and have the same name and different parameter lists, the method is said to be overloaded.

An example =

```
class OverloadExample
```

~~// see last page for example~~

```
{
    public OverloadExampleMethod
```

```
{
```

```
// Constructor with no parameters
```

```
{
```

```
{
    public OverloadExampleMethod (string Value)
```

```
{
```

```
// Constructor with an overload of a string parameter
```

```
{
```

```
{
    public OverloadExampleMethod (bool value)
```

```
{
```

```
// Constructor with an overload of a boolean parameter
```

```
{
```

```
}
```

5

(v) A constructor is called everytime a class or struct is created. If a constructor has not been provided for in the code, C++ creates one by default that instantiates the object to the default values.

A constructor is a method with the same name as the name of the type (class/struct), and does not have a return type.

An example

```
public class MyClass
```

```
{
```

```
    private int firstNumber;
```

```
    private int secondNumber;
```

```
    // Constructor with no values
```

```
    public MyClass ()
```

```
    {
```

```
        firstNumber = 0;
```

```
        secondNumber = 0;
```

```
    }
```

```
    // Constructor overload requiring two ints
```

```
    public MyClass (int value1, int value2)
```

```
    {
```

```
        firstNumber = value1;
```

```
        secondNumber = value2;
```

```
    }
```

```
}
```

2

Overloading is an important technique to improve usability, productivity and reliability.

Overloading on the number of parameters makes it possible to provide simpler versions of constructors and methods.

Overloading on the parameter type makes it possible to create an object with different data types and still create the object successfully.

A simple no value constructor makes it quite easy to create a default object without worrying about the values being available at time of construction. A common scenario is by passing default values to ~~the~~ the complex overloads.

```
class OverloadExample
```

```
{
```

```
    public OverloadExample ()
```

```
    {
```

```
        // constructor with no parameters
```

```
    }
```

```
    public OverloadExample (int Number)
```

```
    {
```

```
        // constructor with one overload of an integer parameter
```

```
    }
```

```
}
```

5

5

③ (a) public class radio

private bool -isSwitchedOn; // either true or false
public bool isSwitchedOn

get { return -isSwitchedOn; }
set { -isSwitchedOn = value; }

private bool -volume; // from 0 to 10
public bool volume

get { return -volume; }
set

{
if (value < 0)

{
-volume = 0; }

~~else if (value > 10)~~
~~{~~

else if (value > 10)
{ -volume = 10; }

else

{
-volume = value;

}

③ (a)
Continued

private double frequency; // from 535 to 1605

public double frequency;

get { return -frequency; }

set

{
if (value < 535.0f)

{
-frequency = 535.0f;

}
else if (value > 1605.0f)

{
-frequency = 1605.0f;

}
else

{
-frequency = value;

}

// set
public double frequency

// public bool volume

③ Continued

(b) Three instance variables, which two?

~~public void SwitchOn~~ (bool isSwitchedOn, int volume)

public radio (bool isSwitchedOn, int volume)

- isSwitchedOn = this.isSwitchedOn;

volume = this.volume;

public radio (bool isSwitchedOn, double frequency)

- isSwitchedOn = this.isSwitchedOn;

~~frequency~~ frequency = this.frequency;

public radio (int volume, double frequency)

Change Frequency And Volume (this.volume, this.frequency)

public void Change Frequency And Volume (int volume, double frequency)

Volume = this.volume;

frequency = this.frequency;

public void IsSwitchedOn (bool value)

- isSwitchedOn = value;

Compile Time Polymorphism is Early Binding / Overloading / Static Binding. This means we declare methods with same name but different signatures because of this we will perform different tasks with same method name.

Example:

Class Polymorphism Example

// Compile time Example

public int numbersAdd (int firstNumber, int secondNumber)

return (firstNumber + secondNumber);

public int numbersAdd (int firstNumber, int secondNumber, int thirdNumber)

return (firstNumber + secondNumber + thirdNumber);

⑥

~~polymorphic~~ Runtime polymorphism means we have the same method names with same signatures. In runtime polymorphism we can override a method in the base class by creating similar functions in a derived class. This can be achieved by using inheritance principle and using virtual and override keywords. This is called Late Binding / Overriding / Dynamic Binding.

Example =

```
public class baseClass
{
    public virtual void sampleMethod()
    {
        Console.WriteLine("Base Class");
    }
}

// Derived Class
public class derivedClass : baseClass
{
    public override void sampleMethod()
    {
        Console.WriteLine("Derived Class");
    }
}
```

Class Runtime Polymorphism Example

```
static void Main (string[] args)
{
    // calling the overridden method
    derivedClass derivedExample = new derivedClass();
    derivedExample.sampleMethod();
}
```

// calling the base class method
baseClass baseExample = new baseClass();
baseExample.sampleMethod();
// static void
// class

① (iv) (continued...)

```
// class MethodOverloadExample
```

```
{
```

```
    public void DoAction()
```

```
    {
```

```
        DoAction(ConvertStringToInt(value));
```

```
    }
```

```
    // Main function to handle the value
```

```
    // ...
```

2

2

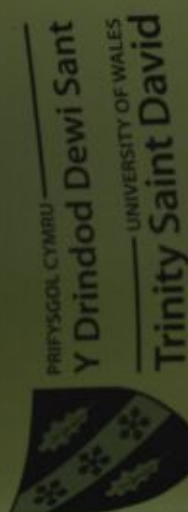
Surname

JONES REES

(Block Capitals) Forename(s)

STEFFAN

Please complete this section and SEAL before the examination starts.

Complete
this section
before the
examination
starts.
FOLD

PRIFYSGOL CYMRU

Y Drindod Dewi Sant

UNIVERSITY OF WALES

Trinity Saint David

Examination Answer Book

12 Pages

Instructions to candidates

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right and seal to hide your name.
3. The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

1	Student Number	95625125	List the numbers of the questions you have attempted	Office Use
2	Title of Examination Paper	Application design and development	1	5
3	Module Code	CSAC5001	2	2
4	Date and Time of Examination	9/10/16 11:30 AM	3a	1
5	Examination Venue	Coleg Sir Gwyn Gwyn Campus	5	6
6	Seat Number	N/A	7	3
7	Number of Answer Books handed in		8	1
8			9	3
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				
46				
47				
48				
49				
50				
51				
52				
53				
54				
55				
56				
57				
58				
59				
60				
61				
62				
63				
64				
65				
66				
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				
101				
102				
103				
104				
105				
106				
107				
108				
109				
110				
111				
112				
113				
114				
115				
116				
117				
118				
119				
120				
121				
122				
123				
124				
125				
126				
127				
128				
129				
130				
131				
132				
133				
134				
135				
136				
137				
138				
139				
140				
141				
142				
143				
144				
145				
146				
147				
148				
149				
150				
151				
152				
153				
154				
155				
156				
157				
158				
159				
160				
161				
162				
163				
164				
165				
166				
167				
168				
169				
170				
171				
172				
173				
174				
175				
176				
177				
178				
179				
180				
181				
182				
183				
184				
185				
186				
187				
188				
189				
190				
191				
192				
193				
194				
195				
196				
197				
198				
199				
200				
201				
202				
203				
204				
205				
206				
207				
208				
209				
210				
211				
212				
213				
214				
215				
216				
217				
218				
219				
220				
221				
222				
223				
224				
225				
226				
227				
228				
229				
230				
231				
232				
233				
234				
235				
236				
237				
238				
239				
240				
241				
242				
243				
244				
245				
246				
247				
248				
249				
250				
251				
252				
253				
254				
255				
256				
257				
258				
259				
260				
261				
262				
263				
264				
265				
266				
267				
268				
269				
270				
271				
272				
273				
274				
275				
276				
277				
278				
279				
280				
281				
282				
283				
284				
285				
286				
287				
288				
289				
290				
291				
292				
293				
294				
295				
296				
297				
298				
299				
300				
301				
302				
303				
304				
305				
306				
307				
308				
309				
310				
311				
312				
313				
314				
315				
316				
317				
318				
319				
320				
321				
322				
323				
324				
325				
326				
327				
328				
329				
330				
331				
332				
333				
334				
335				
336				
337				
338				
339				
340				
341				
342				
343				
344				
345				
346				
347				
348				
349				
350				
351				
352				
353				
354				
355				
356				
357				
358				
359				
360				
361				
362				
363				
364				
365				
366				
367				
368				
369				
370				
371				
372				
373				
374				
375				
376				
377				
378				
379				
380				
381				
382				
383				
384				
385				
386				
387				
388				
389				
390				
391				
392				
393				
394				
395				
396				
397				
398				
399				
400				
401				
402				
403				
404				
405				
406				
407				
408				
409				
410				
411				
412				
413				
414				

1a) A class is a data structure which contains variables and methods within a single entity.

A class declaration comprises a class access

specifier. ~~Example~~ A declaration of a class

is followed by a pair of ~~brackets~~ braces which

contains the variable and method declaration.

1b) An object is a basic principle of building an OOP.

It packs together data and methods to manipulate data.

Objects in an OOP can represent physical data such as

customer details, product details. Objects can also be a computer

implementation feature such as GUI windows, files or linked

lists.

1c) An abstract ~~class~~ ^{Data type} is like a normal ~~class~~ ^{Data type}. It can

be defined just like a normal one. It has fields and

methods which can be inherited from other data types when

a programmer creates a class which implements an interface

a programmer agrees to implement all of the properties

and methods defined in the interface.

1d) If two identifiers have the same name and are

declared within the same scope, they are considered overloaded.

Often an overloaded identifier is a bug which is trapped as

a compile-time error. An example of this is if you declare

two local variables in the same method with the same

name. The compiler reports an error.

1e) When a new keyword is used ~~with~~ to create a new

object, the runtime needs to construct that object by using

the definition of a class. The run time must grab a piece

of memory from the operating system, fill it with fields

defined by the class and invoke a constructor to perform

any initialization required.

2) A constructor is a special kind of method, just like all methods it can be overloaded. As there may be many concise, writeLine methods, each of which takes different

parameters, & a programmer could just write different versions of a constructor. Overloading a constructor is useful when you need to perform the same operation on different data types or varying groups of information. You could overload a constructor when different implementations have different sets of parameters.

5) In polymorphism, an operation can have the same name within different classes and proceed differently in each class. An example of this is you could open a door, open a window, open a newspaper. In each case a different operation is performed. In object orientation, each class knows that operation is supposed to take place.

Example of polymorphism.

Class Management

```
public virtual String GetPageName()
```

• return "This is a Mammal";

Class Horse: Mammal

public override String GetTypeName()

Return "This is a Horse";

7) Actors

visitors
receptionist
customers

- ↓ Visitors book room on website
- ↓ Customer receives reservation number for room
- ↓ Visitor Arrives at the hotel

* Receptionist receives the reservation number

- * Receptionist finds the customers details and takes them to room

- ↳ Receptionist verifies payment through card payment system

* Invoice is provided for the customer if requested

to System prepares monthly reports for Hotel Manager by request

8

Parking Enforcement

New lines: Integer

Parking Attendant: Strong

9)

(i) This indicator means one or many objects in a class

(ii) This indicator means zero or more objects in a class

(iii) This indicator means zero or one objects in a class.

(iv) This indicator means 2 objects in a class

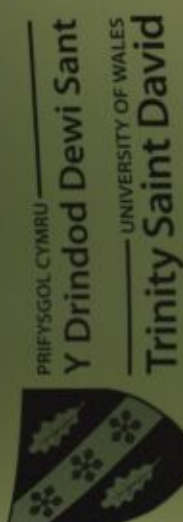
(v) This indicates many objects in a class

3) a) ~~Public class Radio~~
~~Public Radio {~~
~~isSwitchedOn;~~
~~return~~
~~return true;~~
~~else~~
~~return false;~~
~~}~~
~~Public int volume();~~

3a) ~~Public class Radio {~~
~~isSwitchedOn;~~
~~return~~
~~return true;~~
~~else~~
~~return false;~~
~~}~~
~~Public int volume(); { get; set; }~~

Surname **LANG**(Block Capitals) Forename(s) **RYAN PAUL**

Please complete this section and SEAL before the examination starts.



Examination Answer Book

12 Pages

Instructions to candidates

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right and seal to hide your name.
3. The answers must be **legibly** written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

Office Use		
List the numbers of the questions you have attempted	1	9
	2	3
	3	015545
	4	4
	5	7
	6	8
	7	17
	8	0
	9	5
	10	3
Total		9

Student Number	RL157155
Title of Examination Paper	
Module Code	CSAC5001
Date and Time of Examination	09/01/2018 11:30
Examination Venue	
Seat Number	
Number of Answer Books handed in	1

(Gweler yr ochr arall am y fersiwn Cymraeg)

PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

Llyfr Atebion Arholiadau

12 Tudalen

Cyfarwyddiadau i Ymgeiswyr

1. Rhwch yr holl wybodaeth y gofynnir andani yn y blychau isod.
2. Rhwch eich enw yn y lle a ddarperir i'r dde a seliwch i guddio eich enw.
3. Rhaid i'r atebion fod mewn llawysgrifen **ddarllenadwy**. Gall yr Arholwyr naill ai anwybodu sgript amadlenadwy, neu drefnu iddi gael ei theipio a chael yr ymgeisydd i dalu.
4. Ysgrifennwch eich atebion mewn inc.
5. Dechreuwch bob ateb ar dudalen newydd. Gwnewch unrhyw fras nodiadau yn y llyfr atebion a llynnu llinell drwyddynt.
6. Ni ddylid rhwygo unrhyw ran o'r llyfr hwn; rhaid ei gyflwyno yn gyfan.
7. Os defnyddir mwy nag un llyfr, rhwymwch hwy gyda'i gilydd gan ddefnyddio'r tagiau sydd ar gael oddi wrth y gwyllwyr/gwyllwyr.
8. Rhaid dychwelyd pob llyfr. Ni chaniateir mynd â llyfrau atebion, wedi eu defnyddio neu heb eu defnyddio, allan o'r lleoliad arholi o dan unrhyw amgylchiadau.

1	Rhif Myllytiwr	Rhestrwch rifau'r cwestiynau a ymgeiswyd genych	At Ddibenion y Swyddfa
2	Tieitl y Papur Arholiad		
3	Cod y Modwl		
4	Dyddiad ac Amser yr Arholiad		
5	Lleoliad yr Arholiad		
6	Rhif Sedd		
7	Nifer y Llyfrau Atebion a gyflwynwyd	Cyfanswm	

(See the other side for English version)

A wnewch chi lenwi'r rhan hon a'i SELIO cyn i'r arholiad ddechrau.

Cyfenw _____ (LLYTHRENNAU BREISION) Enw(au) cyntaf _____

Question 1

i. A class is like a blueprint for objects. A class consists of variable declarations, properties, and methods. A class is a generalisation of an object (a real thing or concept).

ii. An object is an instance of a class in memory. In C++, the new operator is used to create an instance of the class.

iii. An abstract data type means that it can not be instantiated. Only classes derived from an abstract class can be instantiated. Abstract is used when we want a blueprint of code to be altered to achieve different goals.

iv. Method overloading allows you to define a similar operation in different ways for different data. Overloading happens when multiple methods in a class have the same name, but have different parameters. This is resolved at compile time.

v. A constructor is used to create a default instance of a class. This is a method. We can use it to make default objects.

Question 2

A constructor sets default objects. If we imagine a program for ^{creating} bank accounts, we expect a name and address by default, but what if the address was left blank. We could have different methods with the same name, but ~~it~~ programmed for different values that can return suitable data. So if we have an input of "Rob" and "Address 1" our example can also an input of the parameter "Dave" with an address. They can be passed to different methods using overloading. This is useful because we can have a variety of default objects ~~without~~ without duplicating code.

Question 3.

a. public class radio

```
{
    private boolean isSwitchedOn = false;
    private int[] volume = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    private double frequency = 53.5;
}
```

b. private double frequency = 53.5;

```
public double setFrequency() {
    return frequency;
}

public int setVolume() {
    return volume;
}

public void setFrequency(double frequency) {
    frequency = value;
}
```

Question 4.

If a class depends on another class they are coupled, this is coupling. Coupling is object oriented programming is not good design if you follow the facade pattern or dependency inversion principle. Cohesion - This is the opposite of coupling, the code that is not dependent on other code.

If there is a lot of coupling in your code then updating one part of the program can result in other parts of the program not functioning. With cohesion we can remove dependencies and have code that is easily updatable.



In this example the button depends on lamp and can be reused.



By having cohesion (achieved by an interface)

Question 5.

In compile time polymorphism we declare methods with different signatures but the same name. This is also called early binding or method overloading.

class Example

```
{
    static void Example(int x)
    {
        console.WriteLine("Example (int x)");
    }

    static void Example(string y)
    {
        console.WriteLine("Example (string y)");
    }
}
```

static void Main()

```
{
    Example("text");
}
```

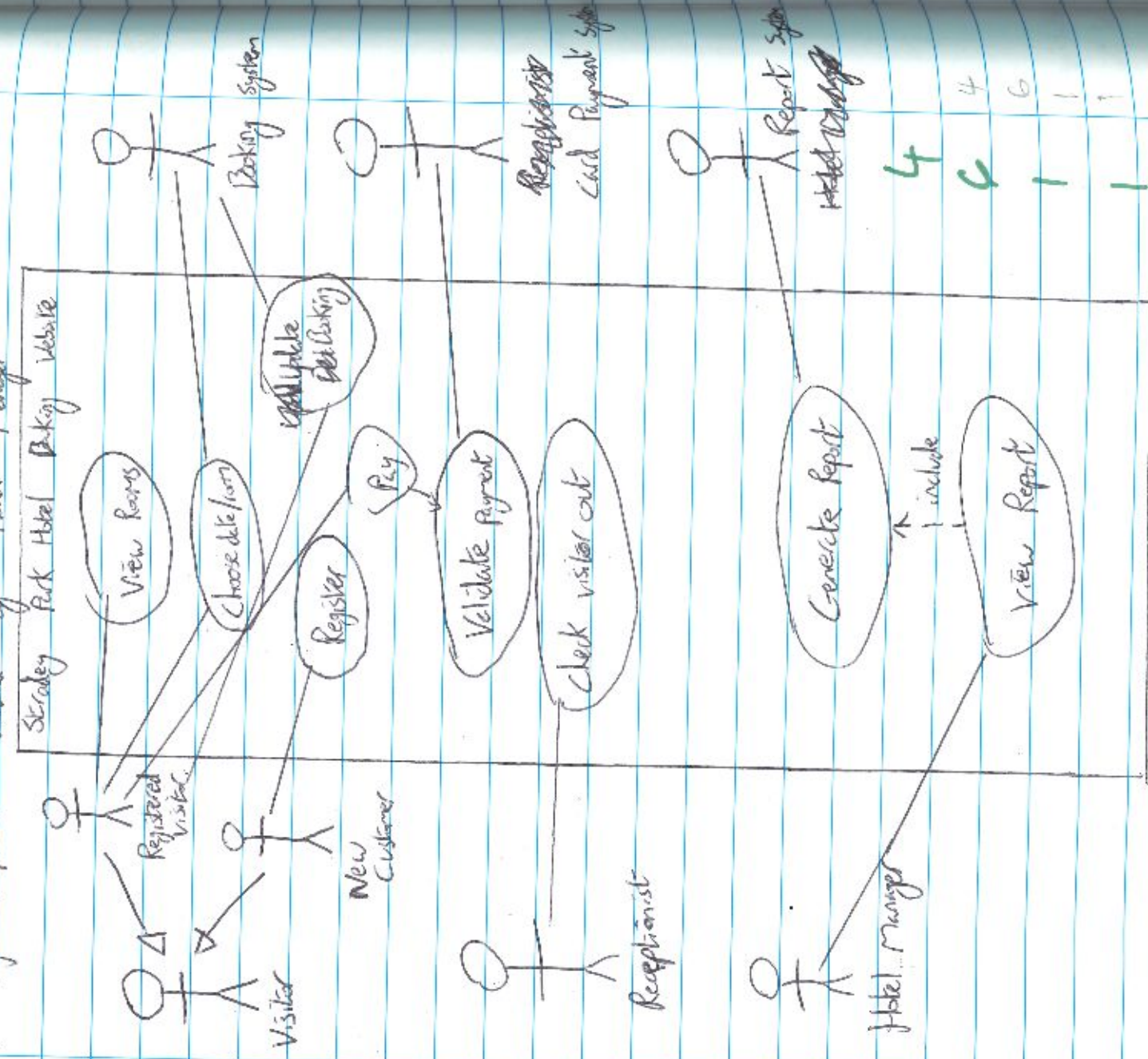
In this example we have methods with the same name and different parameters. This makes code that can handle multiple inputs.

Question 7

Notes Stradley Park Hotel Website

Visitors - Room Reservations (Specify dates / type of room)
 New customer / registered visitor - given unique reservation no.
 Receptionist uses reservation no. checks visitor at
 card payment system - invoice may be requested
 Monthly reports viewed by Hotel manager

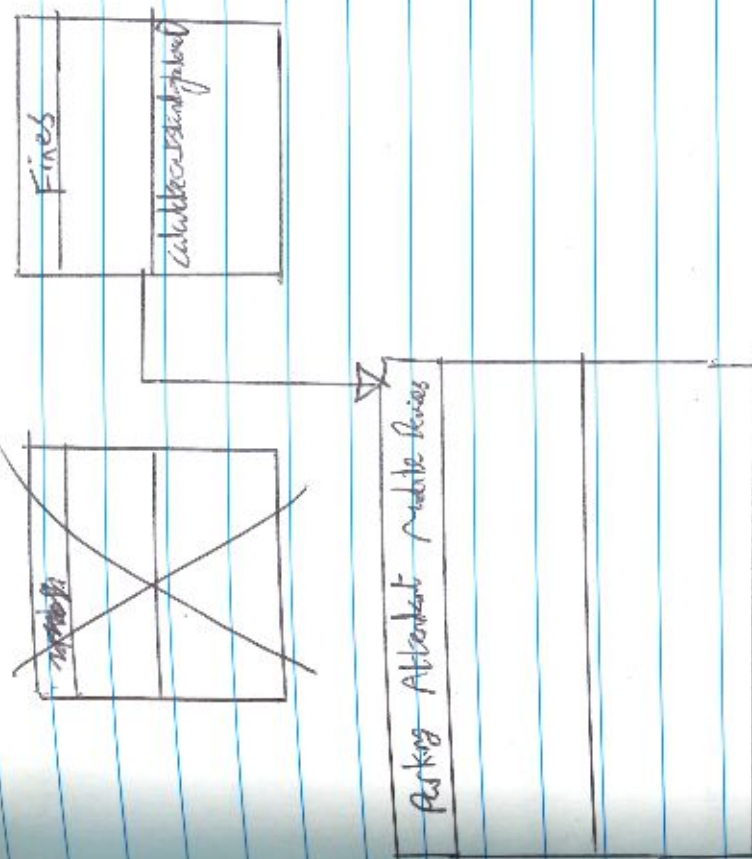
Answer:



4 4 6 1 1

Question 8

LLPES



Question 9.

- 1 or more
- 0 or more
- 0 or 1
- 2 only
- Unlimited (including 0)

Question 10.

- Public
- Private
- Abstract internal
- Reference
- Protected

Question 6.

Run time polymorphism can also be called late binding or method overriding. This uses methods with the same names and signatures, changing them where appropriate in our code.

Class A

```
{
    public void walk()
    {
    }
```

```
    Console.WriteLine("Walk class A");
}
```

Class B: A Inherits from class A

```
{
    new public void walk()
    {
    }
```

```
    Console.WriteLine("Walk class B");
}
```

```
}
```

```
class C
```

```
{
```

```
    public static void main()
```

```
{
```

```
    B objB = new B();
```

```
    objB.walk();
```

```
}
```

```
}
```

This code would output "Walk class B". The original method from class A has been overridden. This happens at run time. Instead of having call repeat we can inherit and override call to make it work in different contexts. Abstract classes require overriding methods.

Complete
section,
remove
stitch, fold as
indicated
and SEAL

FOLD

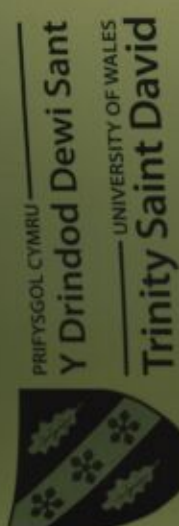
Surname

Newlands NEWLANDS

(Block Capitals) Forename(s)

JOSHUA DANIEL SOHN

Please complete this section and SEAL before the examination starts.



PRIFYSGOL CYMRU

Y Drindod Dewi Sant

UNIVERSITY OF WALES

Trinity Saint David

Examination Answer Book

12 Pages

Instructions to candidates

- Please give the information requested in the boxes below.
- Enter your name in the space provided to the right and seal to hide your name.
- The answers must be **legibly** written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
- Write your answers in ink.
- Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
- No part of this book is to be torn out; it must be handed in complete.
- If more books than one are used, fasten them together using the tags available from the invigilator(s).
- Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

1	Student Number		List the numbers of the questions you have attempted	Office Use
2	Title of Examination Paper	Application Design and development	Q1	8
3	Module Code	CSA55001	Q2	4
4	Date and Time of Examination	09/10/2018	Q3	9
5	Examination Venue		Q4	10
6	Seat Number		Q5	8
7	Number of Answer Books handed in		Q6	13
8			Q7	9
9			Q8	1
10			Q9	5
11			Q10	3
12				
Total				69

(Gweler yr ochr utall am y fersiwn Cymraeg)



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

Llyfr Atebion Arholiadau

12 Tudalen

Cyfarwyddiadau i Ymgeiswyr

- Rhowch yr holl wybodaeth y gofynnir andani yn y blychau isod.
- Rhowch eich enw yn y lle a ddarperir i'r dde a seliwch i guddio eich enw.
- Rhaid i'r atebion fod mewn llawysgrifen **ddarlennadwy**. Gall yr Arholiadau naill ai anwybyddu sgrïp amarlennadwy, neu ddefnu iddi gael ei theipio a chael yr ymgeisydd i dalu.
- Ysgrifennwch eich atebion mewn inc.
- Dechreuwch bob ateb ar dudalen newydd. Gwnewch unrhyw fras nodiadau yn y llyfr atebion a thynnu llinell drwyddyn.
- Ni ddylid rhwygo unrhyw ran o'r llyfr hwn; rhaid ei gyflwyno yn gyfan.
- Os defnyddir mwy nag un llyfr, rhwymwch hwy gyda'i gilydd gan ddefnyddio'r tagiau sydd ar gael oddi wrth y gwyliwr/gwylwyr.
- Rhaid dychwelyd pob llyfr. Ni chaniateir mynd â llyfrau atebion, wedi eu defnyddio neu heb eu defnyddio, allan o'r lleoliad arholi o dan unrhyw amgylchiadau.

1	Rhif Myfyrwr		Rheswch rifau'r cwestiynau a ymgeiswyd gennych	At Ddibenion y Swyddfa
2	Teitl y Papur Arholiad			
3	Cod y Modwl			
4	Dyddiad ac Amser yr Arholiad			
5	Lleoliad yr Arholiad			
6	Rhif Sedd			
7	Nifer y Llyfrau Atebion a gyflwynwyd			Cyfanswm

A wnewch chi lenwi'r rhan hon a'i SELIO cyn i'r arholiad ddechrau.

Cyfenw

(LLYTHRENNAU BREISION) Enw(au) cyntaf

LENWCH Y
RHAN HON,
BYNNEWCH Y
SELYN,
A CHANWCH I
A SELIOGWCH

Section A

Q1

(i) A class describes a real life ^{type of} object, containing the blueprints for the main code to call and create an object of that type. It contains methods, properties, fields, events, delegates and nested classes. An example of a class would be a dog and its features would be properties such as ^{name} colour, breed, height, running speed, etc. In C# you would declare a class like so:

```
class Dog  
{  
}
```

(ii) An object is an instantiation of a class during run time, it is created in the main area of the code to follow the blueprints of a class. Following the previous example of a dog being the class, an object would then possibly be a dog named "Spot" with a colour of brown. To create an object in C# it could be called like so:

```
Dog Spot = new Dog();
```

(iii) Abstract data type is the term given

```

class Dog {
    public String name, colour;
    public Dog (String name, String colour) {

```

```

        name = name;
        colour = colour;
    }
}

```

class main {

```

    Dog dog1 = new Dog("Self", "brown");
}

```

This dog would then be called Self and have the colour brown.

Q2. A constructor can be left blank to allow for a default set of values to be assigned to

the properties of a newly created object, avoiding unnecessary errors when creating code. Using method overloading on a constructor would cut down on code redundancy / code repetition as you would then be able to have in custom arguments into your constructor as you are creating the object rather than setting and resetting those parameters later on in the code. This kind of dynamic coding can also be useful for taking in user input to initialise classes that they make.

(iv) Method overloading is the creation of multiple methods in a class with the same name but different parameters. This is an example of how to achieve compile time polymorphism. An example of this is having a method called add which if used with two numbers results in adding the numbers but if used with two words it simply adds them into one string. In C++ this may be achieved through the following:

```

public int Add (int num1, int num2) {
    return num1 + num2;
}

public String Add (String word1, String word2) {
    return word1 + " " + word2;
}

```

(v) A constructor is a method that is called on the creation of an object to initialise the object, it can accept arguments to change the initialisation from a default to a custom object. An example of this would be creating an object from the class Dog, if no arguments are given then the dog will automatically be given the name 'Spot' with colour brown. In code this may look like this:

Q3

(a) public class radio {

```
public boolean isSwitchedOn; // either true or false.
private int volume; // 0 to 10
private double frequency; // from 535 to 1605 MHz
```

```
public int getVolume() {
    return volume;
}

public void setVolume(int value) {
```

```
    double frequency = 535 + 0.01 * value;
    getFrequency();
    setFrequency(frequency);
}
```

Q4. The term coupling is referring to how closely "nested" the classes are which is in respect to how dependent they are on each other. Having "tightly coupled" classes is where the classes rely on each other for a lot of their methods and should one or the other be missing or changed, it would result in an error. This would not code well for modulation and is generally thought of as bad software design due to this. Better software design is where the classes are loosely coupled possibly with the use of dependency injection using interfaces to decouple the classes.

In the first example of tightly coupled classes an example may look like this

```
public class Class1 {
    public Class2 class2 { get; set; }
}
```

5

5

4

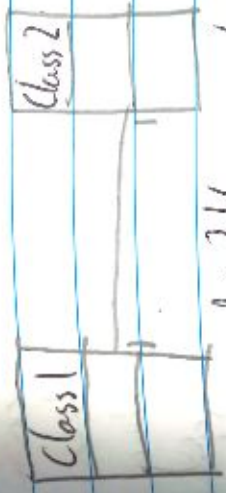
4

10

10

public class Class2 {

In UML class diagrams this would look like this

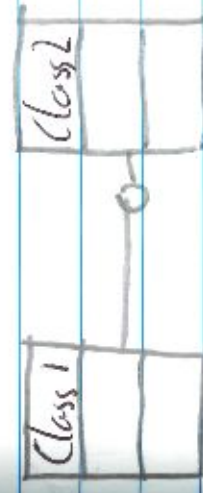


very tightly coupled, no room for any error when the dependency inversion design pattern is implemented the code can change to this

```
public class Class1 {
    public IClass2 class2 { get; set; }
    public interface IClass2 {
        public class Class2 : IClass2 {
            This would then in a class diagram look like this.
```

```
        public class Class2 : IClass2 {
            This would then in a class diagram look like this.
```

This would then in a class diagram look like this.



In this case neither are directly dependent on the other.

Q5 Compile time polymorphism is the use of dynamic use of methods of the same name but different ~~for~~ algorithms at the compile time stage of coding. An example of what would achieve this would be overloading the methods by changing the number of parameters used, changing the order of parameters or using different data types for the parameters.

```

class Dog {
    public String name;
    public Dog() {}
    name = "Jeff";
    public Dog (String name) {
        name = name;
    }
}

class Main {
    Dog dog1 = new Dog();
    Dog dog2 = new Dog("Bob");
}

```

Q6 Run time polymorphism is where there is dynamic use of methods of the same name but different algorithm during runtime. This is achieved by overriding the methods from abstract methods.

```

abstract class Superclass {
    abstract public int Area();
}

class Square extends Superclass {
    int side = 0;
    public Square (int n) { side = n; }
    public override int Area() { return side * side; }
    return side * side;
}

```

Section B

Actor 1 visitor

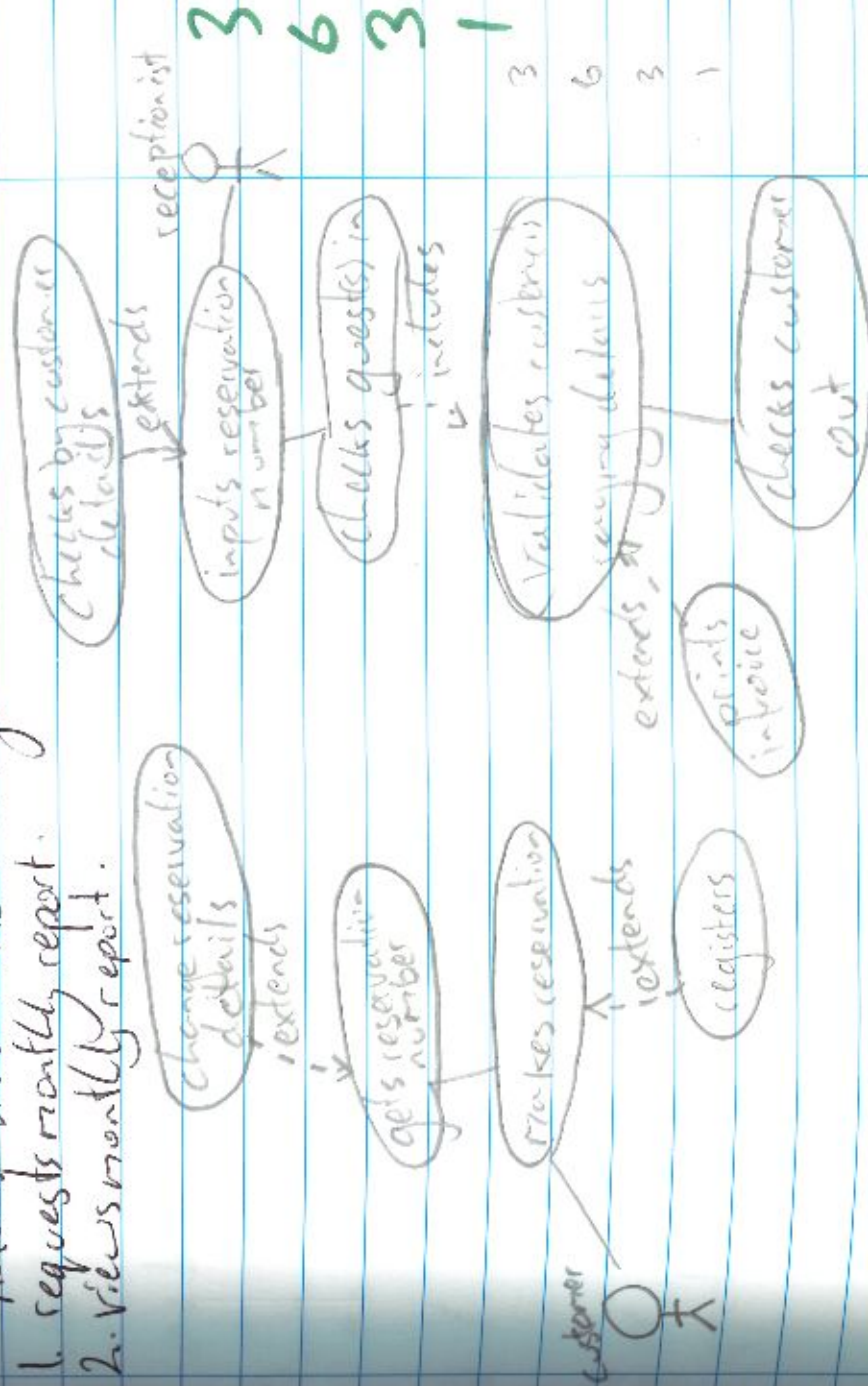
1. Visitor may fragment register.
2. Visitor makes reservation
3. Customer may make changes to their reservation.

Actor 2. Receptionist

1. inputs reservation number / if wrong checks by details.
2. checks customer in
3. Validates payment details
4. may print invoice for customer.
5. checks customer out.

Actor 3. Stradae hotel manager

1. requests monthly report.
2. views monthly report.



manager

Q8. Classes:

- > User (Give person to pay fine.)
- > Employee (Passing attendance to create new fines)
- > Fines (Give actual fines)

Fines

- id: Double
- Paid: Bool
- license plate: String
- Time entered: date

Q9

- (i) 1..* This means that there may be one or many of this.
- (ii) 0..* This means that there may be zero or more of these.
- (iii) 0..1 This means that there may only exist zero or one of these.
- (iv) 2 This means that there may exist 2 and only 2 of these.
- (v) * This means there is no restriction on how many of these may exist.

Q10

- (i) + This is describing that the parameter or method is public/visible everywhere.
- (ii) - This defines the parameter or method as being private (only visible to the class it is in).
- (iii) ~ This defines the parameter or method as being visible only in the package it is in.
- (iv) /
- (v) # This defines the parameter as being protected.

Llyfr Alebion Arholiadau

12 Tudalen

Cyfarwyddiadau i Ymgeiswyr

1. Rhwch yr holl wybodaeth y gofynnir amdani yn y llychau isod.
2. Rhwch eich enw yn y lle a ddarperir i'r dde a seliwch i guddio eich enw.
3. Rhaid i'r atebion fod mewn llawysgrifenn **ddarlennadwy**. Gall yr Arholwyr naill ai anwybyddu sgrifeddiadau darlennadwy, neu drefnu iddi gael ei threipio a chael yr ymgydd i dalu.
4. Ysgrifennwch eich atebion mewn linc.
5. Dechreuwch bob ateb ar dudalen newydd. Gwnewch unrhyw fras nodiadau yn y llyfr atebion a thynnwch llinell drywyddyn.
6. Ni ddylid rhwng unrhyw ran o'r llyfr hwn; rhaid ei gyflwyno yn gyfan.
7. Os defnyddir mwy nag un llyfr, rhwymwch hwy gyda'i gilydd gan ddefnyddio'r tagiau sydd ar gael oddi wrth y gwyllw/gwyllwyr.
8. Rhaid dychwelyd pob llyfr. Ni chaniateir mynd â llyfrau atebion, wedi eu defnyddio neu heb eu defnyddio, allan o'r lleoliad arholi o dan unrhyw amgylchiadau.

1	Rhif Myfyrwr		
2	Teitl y Papur Arholiad		
3	Cod y Modwl		
4	Dyddiad ac Amser yr Arholiad		
5	Lleoliad yr Arholiad		
6	Rhif Sedd		
7	Nifer y Llyfrau Atebion a gyllynywyd		

A wnewch chi lenwi'r rhan hon a'i SELIO cyn i'r arholiad ddechrau.

Cyfenw _____ (LLYTHRENNAU BREISION) Enw(au) cyntaf _____

PATENT PENDING
UNEEZ®

Complete section, remove strip, fold as indicated and SEAL.

GOLD

(Gweler yr ochr arall am y fferbyn Cymraeg.)

41

Surname

THOMAS

(Block Capitals) Forename(s)

GARETH

Please complete this section and SEAL before the examination starts.

1	Student Number	156355	
2	Title of Examination Paper		
3	Module Code	SAE5001	
4	Date and Time of Examination	9/11/18 11:30	
5	Examination Venue		
6	Seat Number		
7	Number of Answer Books handed in		

Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

If more books than one are used, fasten them together using the tags available from the invigilator(s).

6. No part of this book is to be torn out; it must be handed in complete.

Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.

3. The answers must be legibly written. The Examiners can either ignore illegible answers or charge the candidate with writing illegibly.

2. Enter your name in the space provided to the right and seal to hide your name.

Instructions to candidates

12 Pages

Trinity Saint David

PRIFYSGOL CYMRU
Y Drindod Dewi Sant

1. A Class is a like a blueprint, it allows the creation of custom methods, events and variables. It defines the data and behaviour of a type. A class could be used to define the attributes of an instrument, a vehicle or animals.

Existing classes can be used to create and define new classes. This is called inheritance, it is used to modify and reuse existing attributes already found in other classes.

ii. An object is a block of memory that will be defined and configured to the attributes defined by a class. An object is an instance of a class, it will have the same attributes and methods of the parent class. However, objects maintain a separate copy of an ~~attribute~~ attributes, which can be edited to suit the new parameters.

iii. An abstract data type can only be declared within abstract classes. An abstract class can be instantiated. It is an incomplete class that must be implemented into a derived class. An abstract class can only define members, the functionality of other classes. For the sake of inheriting. To declare an abstract class or object, the abstract modifier should be used.

iv Overloading is a concept used to avoid redundant code. Where the same method name is used multiple times but use different set of parameters. The method that gets called during runtime is resolved at compile time, this removes any chance of an error.

It is

v. A constructor is a mechanism used to create instances and objects. C++ includes a default constructor if the programmer doesn't include one in the code.

Overloading refers to the ability to use a single identifier to define multiple methods in a class. Overloading is used as a way to enforce type checking in function calls during compilation. C++ allows operators and methods to be overloaded by defining static members using the operator keyword.

Overloading helps to reduce redundant code however, overloading can become confusing when used too much. It can reduce the readability of the code and the maintainability of code.

3.

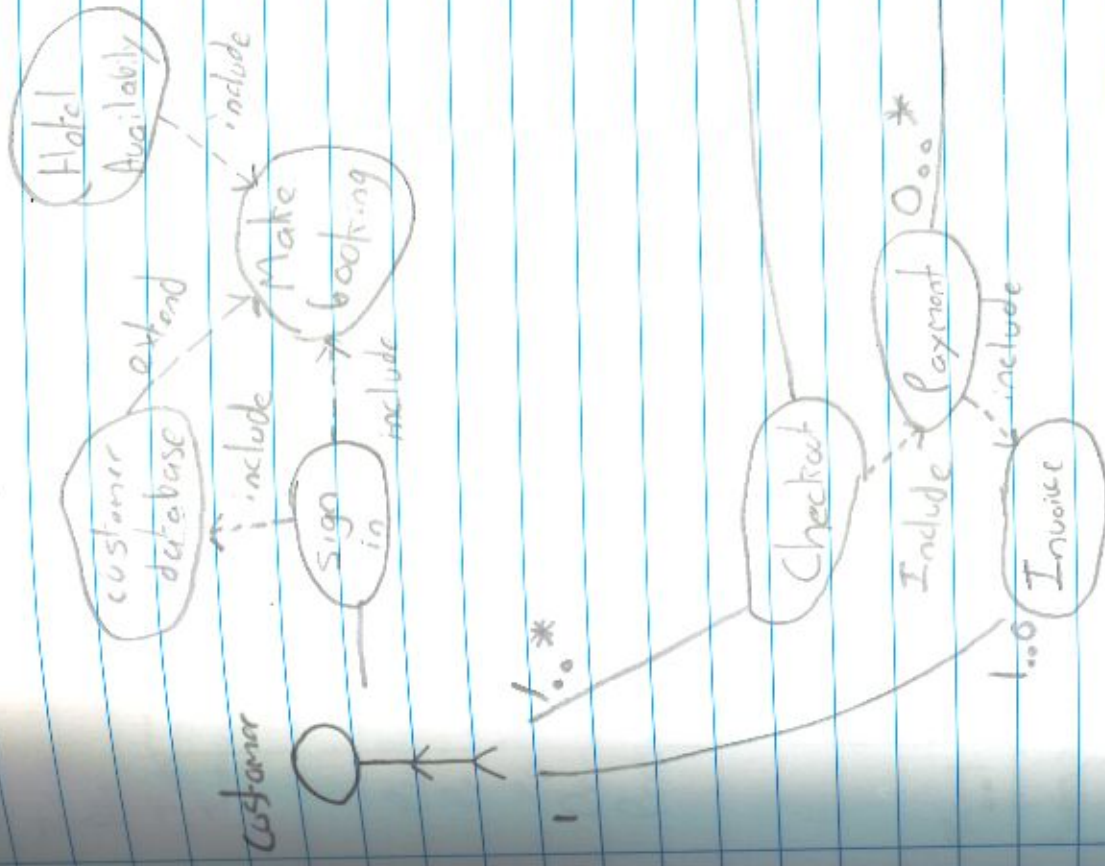
Public Class radio {

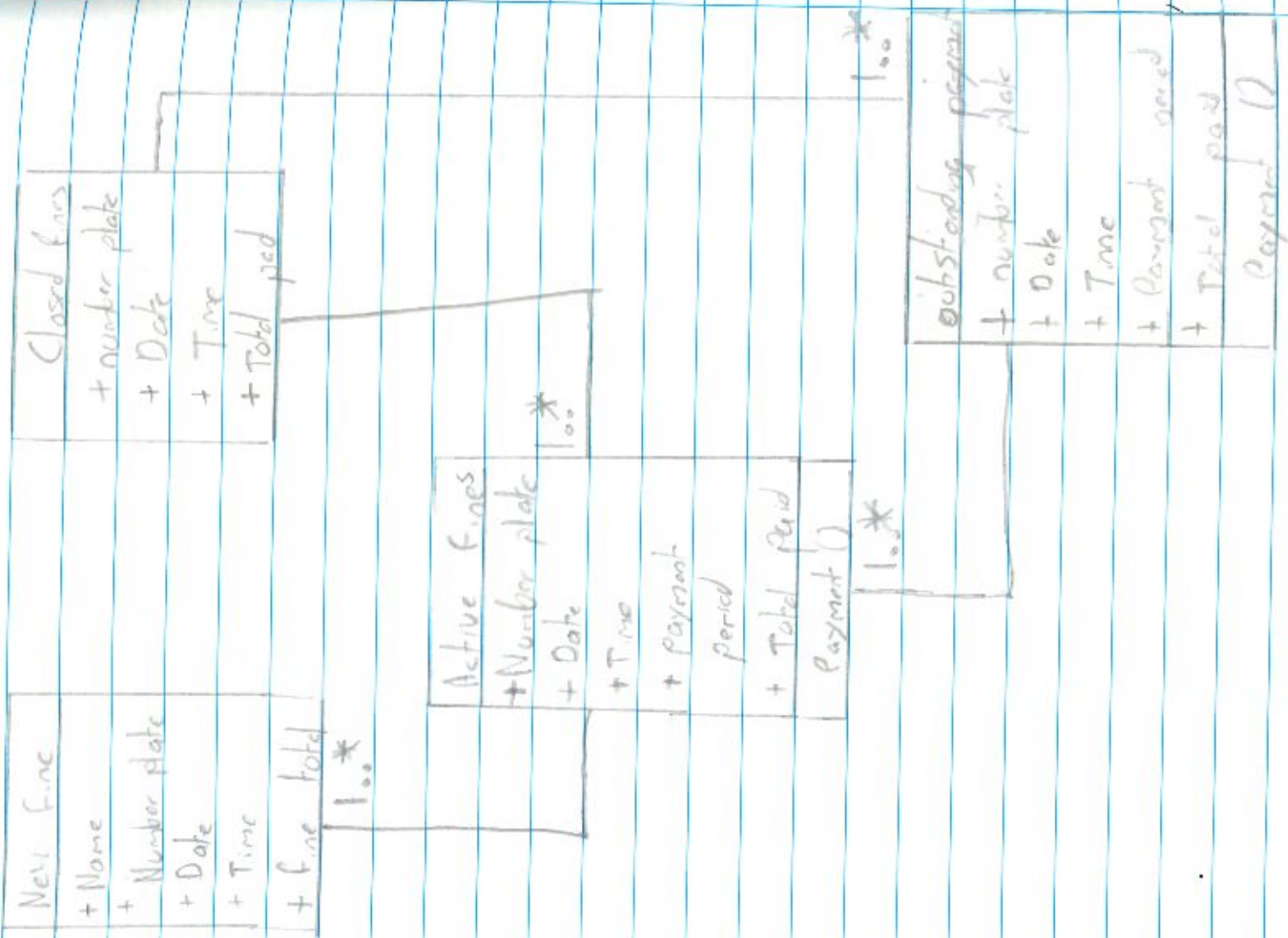
54

1/4

5. Polymorphism is the ability to provide the same interface for different data types.
 A general class will all the attributes and behaviours will be created and shared among different classes.

It allows you to invoke derived class methods through.





- i one or more instances
- ii zero or more instances
- iii No instances, or one instance
- iv Two instances
- v Zero or more instances

- 10: Public
- 11: Private
- 12: Package
- 13: Derived
- 14: Protected

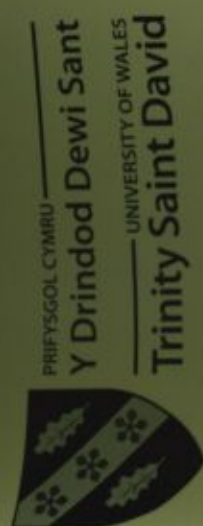
Surname

(Block Capitals) Forename(s)

Please complete this section and SEAL before the examination starts.

Complete this section, remove the paper as instructed and fold as shown.

FOLD



Examination Answer Book

12 Pages

Instructions to candidates

- Please give the information requested in the boxes below.
- Enter your name in the space provided to the right and seal to hide your name.
- The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
- Write your answers in ink.
- Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
- No part of this book is to be torn out; it must be handed in complete.
- If more books than one are used, fasten them together using the tags available from the invigilator(s).
- Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

List the numbers of the questions you have attempted	Office Use
1	8
2	4
3	4+
4	7
5	10
6	12
7	10
8	4
9	5
Total	64

1	Student Number	171460
2	Title of Examination Paper	APPLICATION DESIGN AND DEVELOPMENT
3	Module Code	CSAC5001
4	Date and Time of Examination	11:30, 9/1/18
5	Examination Venue	
9	Seat Number	

7	Number of Answer Books handed in	
---	----------------------------------	--

(Gweler yr ochr arall am y fersiwn Cymraeg)



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

Llyfr Atebion Arholiadau

12 Tudalen

Cyfarwyddiadau i Ymgeiswyr

- Rhowch yr holl wybodaeth y gofynnir amdani yn y blychau isod.
- Rhowch eich enw yn y lle a ddarperir i'r dde a seliwch i guddio eich enw.
- Rhaid i'r atebion fod newn llawysgriflen ddarlennadwy. Gall yr Arholwyr natll ai anwybyddu sgript annarlennadwy, neu ddefnyddio gael i helpio a chael yr ymgeisydd i dalu.
- Ysgrifennwch eich atebion newn inc.
- Dechreuwch bob ateb ar dudalen newydd. Gwnewch unrhyw fras nodiadau yn y llyfr atebion a thynnu llinell drwyddyn.
- Ni ddylid rhwygo unrhyw ran o'r llyfr hwn; rhaid ei gyflwyno yn gyfan.
- Os defnyddir mwy nag un llyfr, rhwymwch hwy gyda'i gilydd gan ddefnyddio'r tagiau sydd ar gael oddi wrth y gwyllw/gwyllwyr.
- Rhaid dychwelyd pob llyfr. Ni chaniateir mynd â llyfrau atebion, wedi eu defnyddio neu heb eu defnyddio, allan o'r lleoliad arholi o dan unrhyw amgylchiadau.

1	Rhif Myfyriwr		Rheswch rîâu'r cwestiynau a ymgeisiwyd gennych	At Ddbecion y Swyddia
2	Tecil y Papur Arholiad			
3	Cod y Modwl			
4	Dyddiad ac Amser yr Arholiad			
5	Lleoliad yr Arholiad			
6	Rhif Sedd			
7	Nifer y Llyfrau Atebion a gyflwynwyd			Cyfanswm

A wnewch chi lenwi'r rhan hon a'i SELIO cyn i'r arholiad ddechrau.

Cyfenw

(LLYTHRENNAU BREISION) Enw(au) cyntaf

Section A

Question 1

1) A class is a template that is used to create and define objects, methods and other data members. A simple explanation of a class would be a blueprint containing design instructions.

An example of a class in code would be:

```
class Employee
{
    private long employeeid;
```

This class has one member known as `employeeid` which has a access modifier of `private`.

11) Objects can be compared to any real world object, such as a pen, computer, book and anything else in between. Objects can be stored within variables, arrays or in Collections.

```
Car = new Car("Red");
Console.WriteLine(car.Describe());
```

Here, `Car` is the object in question. The `car` object is a part of a larger program, and is within a class.

III) Data Types are various forms of data. How data can be represented in a program. Bool (true or false), int (a data type for storing numbers without decimals), string (storing text) and float (storing numbers that could have decimals) are all examples of data types.

Abstract data types, like abstract classes and abstract methods, cannot be implemented.

IV) Method overloading is when a program can use multiple methods with the same name; the actual purpose and arguments of these methods is different.

An example of this is a registration program that needs different information. The different methods will have different names that can explain what they do, but method overloading can change this so their names are the same.

```
public void writeEntry (string)
```

```
{  
}
```

```
public void writeEntry (int password)
```

Using ~~new~~ method overloading, a user can simply implement the two methods quickly because the differences are minimal, instead of having to write out ~~the~~ different names.

V) A constructor is created whenever an ~~instance~~ instance of a class is created. They allow users to ~~perform~~ perform a 'set up' of the class' instance. Constructors have the same name of their containing class always.

```
class Plane
```

```
{  
    public Plane (string newIdentifier)
```

```
{  
    Identifier = newIdentifier  
}
```

```
    public string Identifier
```

```
{  
    get?  
    private set?  
}
```

Here, the constructor can use ~~the~~ the Plane's Identifier. Also, because the constructor is a member of the class, it can use the Identifier's private ~~setter~~ setter.

Question 2

Constructor overloading follows the same principle as method overloading. If a multiple Constructors are named after one class, then they have different information but enough similarities that don't warrant much name change, then it can be beneficial to keep their names the same.

Question 3

Public class Radio

{
 Private bool isSwitchedOn // true or false
 Private int volume; // 0 to 10
 Private double frequency; // from 535 to 1605 MHz
}

Question 4

Coupling is the term of when something is ~~couple~~ coupled to something else: a class coupled to another class, a function to a object, or a object to a class.

The idea of cohesion stems from structured design. Cohesion measures the degree of ~~close~~ connectivity among the elements of a module, class or object.

Coupling can lead to great benefits among programs, mainly minimizing the amount of coupling ~~to~~ between two types and minimizing the ~~at~~ number of types between a class or an object for example.

Bad cohesion for example is keeping unrelated abstractions together, like dogs and planes. Good cohesion will be keeping elements of a class or module together.

Question 5

Compile time polymorphism is another name for method overloading, where multiple methods have ^{the same} names with different data ~~types~~ types or parameters. The methods can only be stored within the same class though.

```
class cls Calculate
{
    public static int Square(int number)
    {
        Console.WriteLine("Integer square calc")
        return number * number
    }

    public static double Square(double num)
    {
        Console.WriteLine("double Square calculated")
        return number * number
    }
}
```

Method overloading ~~helps~~ helps with readability of the code overall. Each version of the method will perform the same general function using different data types. In the example, both methods are chosen from during the compile time of the program as to ~~the~~ which one is better suited.

Question 6

Run time polymorphism, or method overriding is when methods that have similar the same names exist, but in different classes, namely ~~base~~ base and derived classes.

```
class BC
{
    public void display()
    {
        System.Console.WriteLine("BC::Display");
    }
}
```

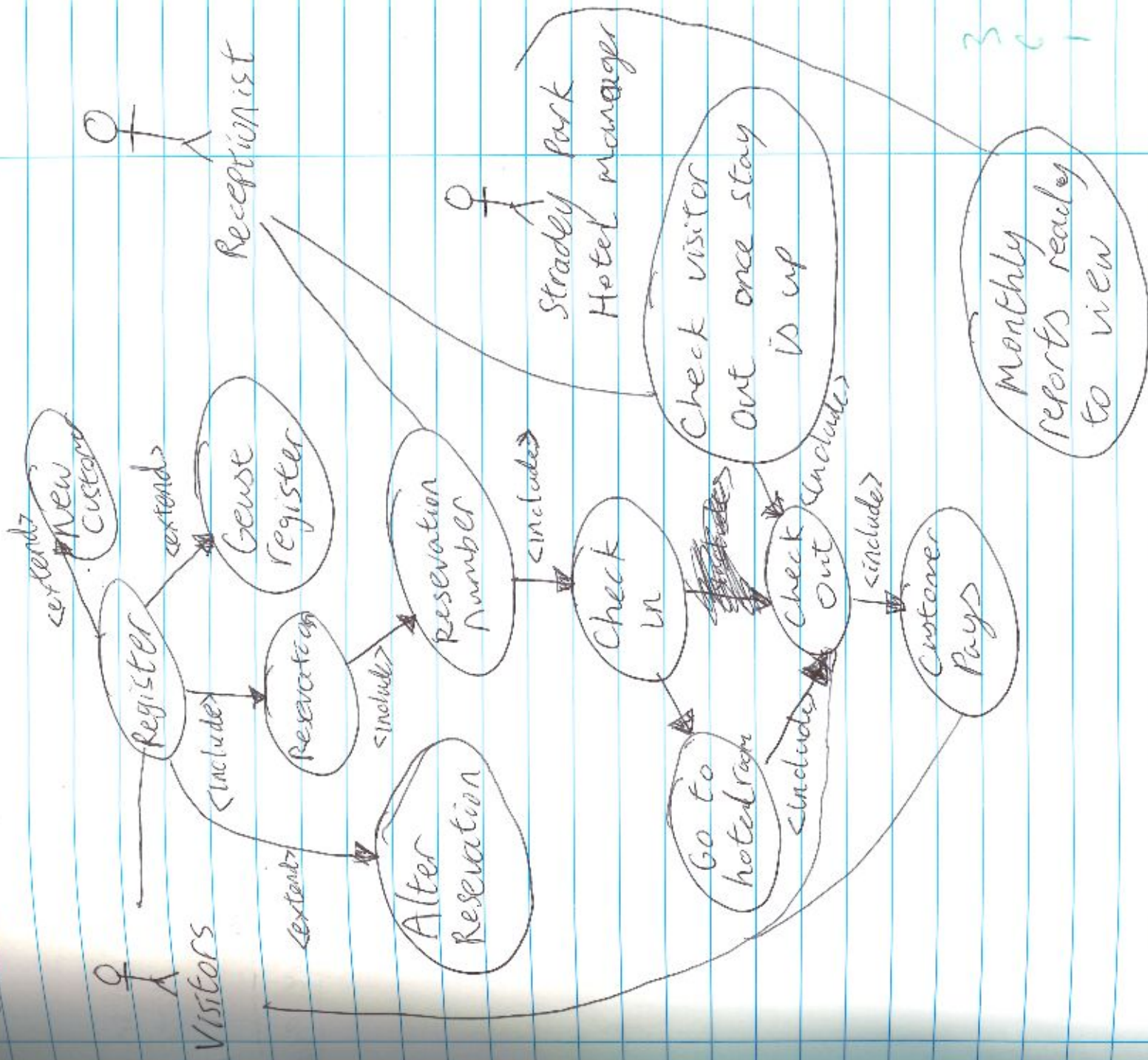
```
class DC: BC
{
    new public void display()
    {
        System.Console.WriteLine("DC::Display");
    }
}
```

```
class Demo
{
    static public void Main()
    {
        BC b;
        b = new BC();
        b.Display();
    }
}
```

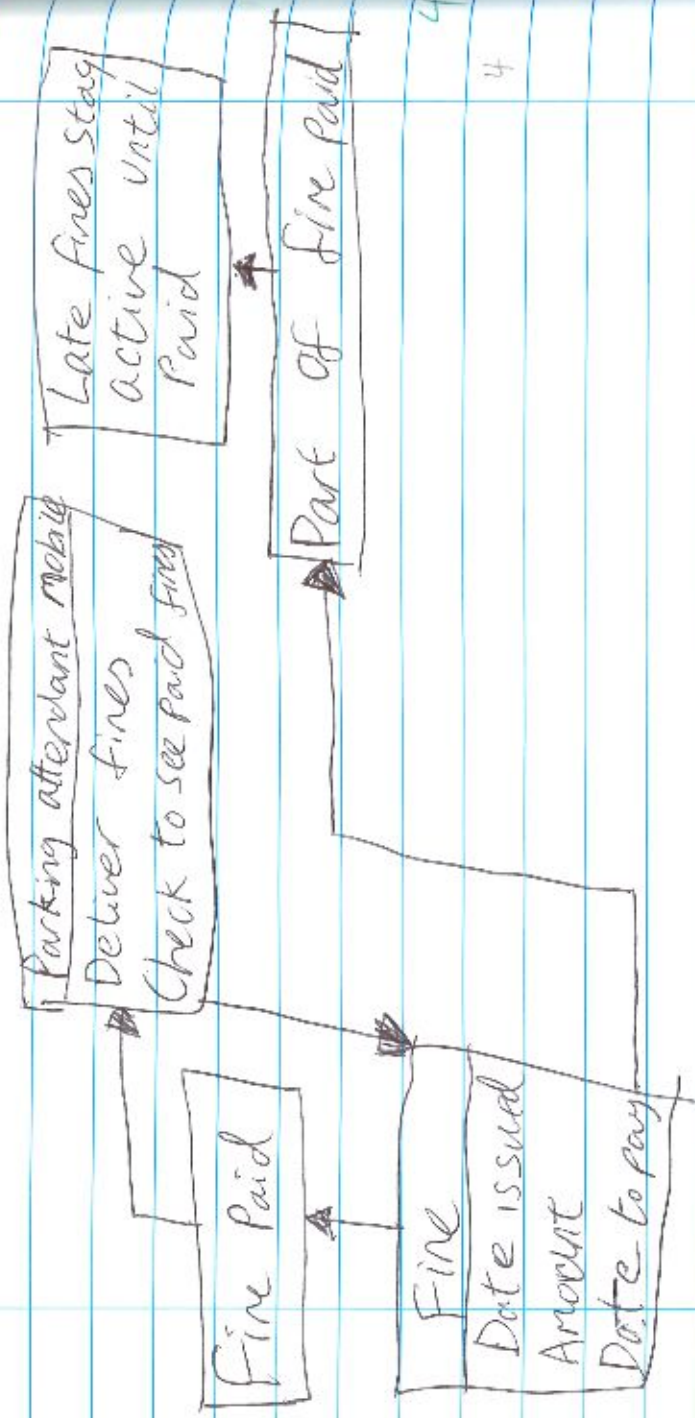
In that example, the Demo class overrides the function of the DC class by making the program output "BC::Display".

If Demo class was not there, the program would output "DC::Display". The 'run time' term comes from the fact that the program reads the code and sees that the Demo class is wanting a new and different output at the end of the programs run.

Question 7



Question 8



Question 9

- i) $1 \dots * = \text{One to Many}$
- ii) $0 \dots * = \text{Zero to Many}$
- iii) $0 \dots 1 = \text{Zero to One}$
- iv) $2 = \text{Exactly Two}$
- v) $* = \text{Many}$