

Practical Assignment 02 (PA02)

	Section A: Abstract Factory Method																						
	Merchandise Hierarchy										Abstract Factory Pattern					Main			UML Class Diagram				
	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
	Merchandise class has correct constructor, taking club, price and type, as well as virtual getDescription function and static id counter	Shirt class constructor sets size and then calls Merchandise constructor with "Shirt" as type. getDescription function calls and extends Merchandise.getDescription function with size	Merchandise class constructor sets inflated price and then calls Merchandise constructor with "SoccerBall" as type. getDescription function calls and extends Merchandise.getDescription function with inflated variable.	LiverpoolShirt constructor takes string variable, and calls Shirt constructor with "Liverpool", any price and the string variable.	ChelseaShirt constructor takes string variable, and calls Shirt constructor with "Chelsea", any price and the string variable.	ArsenalShirt constructor takes string variable, and calls Shirt constructor with "Arsenal", any price and the string variable.	LiverpoolSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Liverpool", any price and the bool variable.	ChelseaSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Chelsea", any price and the bool variable.	ArsenalSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Arsenal", any price and the bool variable.	Total	MerchandiseFactory has pure virtual functions to create a SoccerBall Merchandise item and a Shirt Merchandise item.	ChelseaFactory implements functions to create a ChelseaSoccerBall Merchandise item and a ChelseaShirt, using their created constructors.	LiverpoolFactory implements functions to create a LiverpoolSoccerBall Merchandise item and a LiverpoolShirt, using their created constructors.	ArsenalFactory implements functions to create a ArsenalSoccerBall Merchandise item and a ArsenalShirt, using their created constructors.	Total	ChelseaFactory, LiverpoolFactory and ArsenalFactory are created.	Student asks user which products should be created, as well as options specific to that type of item.	Student creates concrete products using concrete creators, and outputs details of products created.	Total	Correct inheritance in the Merchandise hierarchy, between Merchandise and its child classes (SoccerBall and Shirt) and between these child classes and the concreteProducts.	Correct inheritance between concreteCreators and Creator (e.g. Factory and ChelseaFactory)	Correct associations between concreteCreators and concreteProducts (e.g. ChelseaFactory has associations to ChelseaSoccerBall and ChelseaShirt)	Total
u04483716	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8
u04515146	2	2	2	2	2	2	2	2	2	18	2	0	0	0	2	0	0	0	0	6	0	0	6
u04534205	1	2	2	2	2	2	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0
u15223893	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u15231748	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	0	5	8	5	3	6	14
u16068344	1	1	1	2	2	2	2	2	2	15	2	2	2	2	8	3	3	5	11	5	3	6	14
u16320965	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	0	0	0	0	0	0	0	0
u17016534	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u17029377	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8
u17030553	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u17053928	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u17066736	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u17102210	0	2	2	2	2	2	2	2	2	16	2	0	0	0	2	3	2	2	7	5	3	0	8
u17110310	0	0	0	2	2	2	2	2	2	12	2	2	2	2	8	3	2	5	10	5	3	6	14
u17169811	0	2	1	2	2	2	2	2	2	15	2	2	2	2	8	3	2	5	10	5	3	6	14
u17229457	2	1	2	2	2	2	2	2	2	17	2	2	2	2	8	3	2	5	10	5	3	0	8
u18019499	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u18025685	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8
u18034332	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	0	0	0	0	5	3	0	8
u18053239	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u18055215	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	3	8	3	3	0	6
u18055461	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u18061908	2	1	1	2	2	2	2	2	2	16	2	2	2	2	8	3	2	5	10	5	3	6	14
u18069704	2	2	2	2	2	2	2	2	2	18	0	0	0	0	0	0	0	0	0	5	3	6	14
u18074074	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0
u18080368	2	2	2	2	2	2	2	1	1	16	2	2	2	2	8	0	0	0	0	5	3	6	14
u18105883	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u18108467	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8
u18169253	0	2	2	2	2	2	2	2	2	16	2	2	2	2	8	3	2	5	10	5	3	0	8
u18191135	1	2	2	2	2	2	2	2	2	17	2	2	2	2	8	3	2	5	10	5	3	6	14
u18196366	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	2	10
u18219919	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u18265163	2	0	0	2	2	2	2	2	2	14	2	2	2	2	8	3	2	5	10	5	3	2	10
u18286250	0	2	2	2	2	2	2	2	2	16	2	2	2	2	8	3	2	5	10	5	3	6	14
u18312374	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	3	11

Practical Assignment 02 (PA02)

Section A: Abstract Factory Method																								
	Merchandise Hierarchy										Abstract Factory Pattern					Main				UML Class Diagram				
	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
	Merchandise class has correct constructor, taking club, price and type, as well as virtual getDescription function and static id counter	Shirt class constructor sets size and then calls Merchandise constructor with "Shirt" as type. getDescription function calls and extends Merchandise.getDescription function with size	Merchandise class constructor sets inflated and then calls Merchandise constructor with "SoccerBall" as type. getDescription function calls and extends Merchandise.getDescription function with inflated variable.	LiverpoolShirt constructor takes string variable, and calls Shirt constructor with "Liverpool", any price and the string variable.	ChelseaShirt constructor takes string variable, and calls Shirt constructor with "Chelsea", any price and the string variable.	ArsenalShirt constructor takes string variable, and calls Shirt constructor with "Arsenal", any price and the string variable.	LiverpoolSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Liverpool", any price and the bool variable.	ChelseaSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Chelsea", any price and the bool variable.	ArsenalSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Arsenal", any price and the bool variable.	Total	MerchandiseFactory has pure virtual functions to create a SoccerBall Merchandise item and a Shirt Merchandise item.	ChelseaFactory implements functions to create a ChelseaSoccerBall Merchandise item and a ChelseaShirt, using their created constructors.	LiverpoolFactory implements functions to create a LiverpoolSoccerBall Merchandise item and a LiverpoolShirt, using their created constructors.	ArsenalFactory implements functions to create a ArsenalSoccerBall Merchandise item and a ArsenalShirt, using their created constructors.	Total	ChelseaFactory, LiverpoolFactory and ArsenalFactory are created.	Student asks user which products should be created, as well as options specific to that type of item.	Student creates concrete products using concrete creators, and outputs details of products created.	Total	Correct inheritance in the Merchandise hierarchy, between Merchandise and its child classes (SoccerBall and Shirt) and between these child classes and the concreteProducts.	Correct inheritance between concreteCreators and Creator (e.g. Factory and ChelseaFactory)	Correct associations between concreteCreators and concreteProducts (e.g. ChelseaFactory has associations to ChelseaSoccerBall and ChelseaShirt)		Total
u18335412	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	1	9
u18350110	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	2	10
u18371435	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	2	0	2	4	5	3	6		14
u19004232	2	1	1	2	2	2	2	2	2	16	2	2	2	2	8	3	2	5		10	5	3	6	14
u19009977	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19015951	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	3	3	6	12
u19021209	2	2	2	2	2	1	1	1	1	14	2	2	2	2	8	3	2	5		10	3	3	4	10
u19027372	2	0	0	1	1	1	1	1	1	8	2	2	2	2	8	3	2	5	10	0	0	0	0	0
u19028815	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	5
u19029242	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19033347	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19037717	2	1	1	2	2	2	2	2	2	16	2	2	2	2	8	3	2	5		10	5	3	6	14
u19044233	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	0	8
u19048280	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	0	8
u19050993	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	2	2	4	8	5	3	0	8	
u19053313	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19061359	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19062665	2	2	1	1	1	1	1	1	1	11	2	2	2	2	8	3	2	5		10	5	3	6	14
u19068035	2	2	1	1	1	1	1	1	1	11	2	2	2	2	8	3	2	5		10	5	3	6	14
u19068710	2	2	1	1	1	1	1	1	1	11	2	2	2	2	8	3	2	5		10	5	3	6	14
u19068752	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19072067										0					0				0					0
u19083786	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19100133	2	2	2	2	2	2	2	2	2	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u19103345	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19116498	2	2	2	1	1	1	1	1	1	12	2	2	2	2	8	3	2	5		10	5	3	6	14
u19123460	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19126353	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19130938	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19141859	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19151952										0					0				0					0
u19153113	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19185032	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14
u19185678	2	2	2	2	2	2	2	2	2		2	2	2	2		3	2	5			5	3	0	
u19198958	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5		10	5	3	6	14

Practical Assignment 02 (PA02)

	Section A: Abstract Factory Method																							
	Merchandise Hierarchy										Abstract Factory Pattern					Main				UML Class Diagram				
	2	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
	Merchandise class has correct constructor, taking club, price and type, as well as virtual getDescription function and static id counter	Shirt class constructor sets size and then calls Merchandise constructor with "Shirt" as type. getDescription function calls and extends Merchandise class constructor with size	Merchandise class constructor sets inflated and then calls Merchandise constructor with "SoccerBall" as type. getDescription function calls and extends Merchandise class constructor with inflated variable.	LiverpoolShirt constructor takes string variable, and calls Shirt constructor with "Liverpool", any price and the string variable.	ChelseaShirt constructor takes string variable, and calls Shirt constructor with "Chelsea", any price and the string variable.	ArsenalShirt constructor takes string variable, and calls Shirt constructor with "Arsenal", any price and the string variable.	LiverpoolSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Liverpool", any price and the bool variable.	ChelseaSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Chelsea", any price and the bool variable.	ArsenalSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Arsenal", any price and the bool variable.	Total	MerchandiseFactory has pure virtual functions to create a SoccerBall Merchandise item and a Shirt Merchandise item.	ChelseaFactory implements functions to create a ChelseaSoccerBall Merchandise item and a ChelseaShirt, using their created constructors.	LiverpoolFactory implements functions to create a LiverpoolSoccerBall Merchandise item and a LiverpoolShirt, using their created constructors.	ArsenalFactory implements functions to create a ArsenalSoccerBall Merchandise item and a ArsenalShirt, using their created constructors.	Total	ChelseaFactory, LiverpoolFactory and ArsenalFactory are created.	Student asks user which products should be created, as well as options specific to that type of item.	Student creates concrete products using concrete creators, and outputs details of products created.	Total	Correct inheritance in the Merchandise hierarchy, between Merchandise and its child classes (SoccerBall and Shirt) and between these child classes and the concreteProducts.	Correct inheritance between concreteCreators and Creator (e.g. Factory and ChelseaFactory)	Correct associations between concreteCreators and concreteProducts (e.g. ChelseaFactory has associations to ChelseaSoccerBall and ChelseaShirt)	Total	
u19214597	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	0	5	8	5	0	0	5	
u19228882	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8	
u19236183	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u19236272	2	2	2	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	
u19290498	2	2	2	2	2	2	2	2	2	18	0	2	2	2	6	3	2	5	10	5	3	6	14	
u19294418	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u19367211	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u20424622	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	3	8	5	3	6	14	
u20426799	2	0	0	1	1	1	0	0	0	5	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20426918	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20428082	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20430168	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	0	2	5	7	5	3	6	14	
u20430516										0					0				0				0	
u20430630	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20431997	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20432748	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20435780	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u20435992	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20436077	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
u20437863	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	0	5	8	5	3	6	14	
u20438151										0					0				0				0	
u20439963	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	2	10	
u20440562	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20441135	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20442018	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20443260	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u20443626	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	4	12	
u20444738	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	5	13	
u20448474	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u20450533	2	1	2	2	2	2	2	2	2	17	2	2	2	2	8	3	2	5	10	5	3	5	13	
u20450932	1	2	2	2	2	2	2	2	2	17	2	2	2	2	8	3	2	2	7	5	3	0	8	
u20453222	2	1	2	2	2	2	2	2	2	17	2	2	2	2	8	3	2	5	10	5	3	5	13	
u20453478	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8	
u20456078	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u20460067	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	

Practical Assignment 02 (PA02)

	Section A: Abstract Factory Method																							
	Merchandise Hierarchy										Abstract Factory Pattern					Main				UML Class Diagram				
	2	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
	Merchandise class has correct constructor, taking club, price and type, as well as virtual getDescription function and static id counter	Shirt class constructor sets size and then calls Merchandise constructor with "Shirt" as type. getDescription function calls and extends Merchandise class constructor with size	Merchandise class constructor sets inflated price and then calls Merchandise constructor with "SoccerBall" as type. getDescription function calls and extends Merchandise class constructor with inflated price	LiverpoolShirt constructor takes string variable, and calls Shirt constructor with "Liverpool", any price and the string variable.	ChelseaShirt constructor takes string variable, and calls Shirt constructor with "Chelsea", any price and the string variable.	ArsenalShirt constructor takes string variable, and calls Shirt constructor with "Arsenal", any price and the string variable.	LiverpoolSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Liverpool", any price and the bool variable.	ChelseaSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Chelsea", any price and the bool variable.	ArsenalSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Arsenal", any price and the bool variable.	Total	MerchandiseFactory has pure virtual functions to create a SoccerBall Merchandise item and a Shirt Merchandise item.	ChelseaFactory implements functions to create a ChelseaSoccerBall Merchandise item and a ChelseaShirt, using their created constructors.	LiverpoolFactory implements functions to create a LiverpoolSoccerBall Merchandise item and a LiverpoolShirt, using their created constructors.	ArsenalFactory implements functions to create a ArsenalSoccerBall Merchandise item and a ArsenalShirt, using their created constructors.	Total	ChelseaFactory, LiverpoolFactory and ArsenalFactory are created.	Student asks user which products should be created, as well as options specific to that type of item.	Student creates concrete products using concrete creators, and outputs details of products created.	Total	Correct inheritance in the Merchandise hierarchy, between Merchandise and its child classes (SoccerBall and Shirt) and between these child classes and the concreteProducts.	Correct inheritance between concreteCreators and Creator (e.g. Factory and ChelseaFactory)	Correct associations between concreteCreators and concreteProducts (e.g. ChelseaFactory has associations to ChelseaSoccerBall and ChelseaShirt)	Total	
u20460687	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	0	
u20463163	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20465026	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20466570	2	2	0	2	2	2	2	2	2	16	2	2	2	2	8	0	0	0	0	0	0	0	0	
u20468203	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20469366	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8	
u20471582	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	0	0	0	0	5	3	6	14	
u20478144	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	5	3	6	14	
u20479884	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20481218	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	0	0	0	0	5	3	6	14	
u20486783	2	1	1	1	1	1	1	1	1	10	2	1	1	1	5	0	0	0	0	5	3	6	14	
u20491141	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20494166	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20494654	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	4	2	6	12	
u20498510	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20502126	2	1	1	2	2	2	2	2	2	16	2	0	0	0	2	0	2	0	2	5	0	0	5	
u20504552	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	4	3	6	13	
u20506237	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20507102	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	2	3	0	5	
u20513667	1	1	1	2	2	2	2	2	2	15	2	2	2	2	8	0	0	0	0	5	3	0	8	
u20519517	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20522623	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	0	0	0	0	
u20528036	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20528834	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	0	0	0	0	5	3	6	14	
u20529440	2	2	2	2	2	2	2	2	2	18	1	2	2	2	7	0	0	0	0	5	3	6	14	
u20532581	1	2	2	2	2	2	2	2	2	17	2	2	2	2	8	3	2	2.5	7.5	5	3	0	8	
u20534541	2	2	2	2	2	2	2	2	2	18	1	2	2	2	7	3	2	5	10	5	3	6	14	
u20536951	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8	
u20538945	1	2	2	2	2	2	2	2	2	17	2	2	2	2	8	0	0	0	0	5	3	6	14	
u20554240	2	2	2	2	2	2	2	2	2	18	1	2	2	2	7	0	0	0	0	5	3	6	14	
u20556455	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20557622	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20573783	2	2	2	2	2	2	2	2	2	18	0	2	2	2	6	3	2	5	10	5	3	0	8	
u20575085	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14	
u20578688	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8	

Practical Assignment 02 (PA02)

	Section A: Abstract Factory Method																						
	Merchandise Hierarchy										Abstract Factory Pattern					Main			UML Class Diagram				
	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
	Merchandise class has correct constructor, taking club, price and type, as well as virtual getDescription function and static id counter	Shirt class constructor sets size and then calls Merchandise constructor with "Shirt" as type. getDescription function calls and extends Merchandise constructor with size	Merchandise class constructor sets inflated price and then calls Merchandise constructor with "SoccerBall" as type. getDescription function calls and extends Merchandise constructor with inflated price	LiverpoolShirt constructor takes string variable, and calls Shirt constructor with "Liverpool", any price and the string variable.	ChelseaShirt constructor takes string variable, and calls Shirt constructor with "Chelsea", any price and the string variable.	ArsenalShirt constructor takes string variable, and calls Shirt constructor with "Arsenal", any price and the string variable.	LiverpoolSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Liverpool", any price and the bool variable.	ChelseaSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Chelsea", any price and the bool variable.	ArsenalSoccerBall constructor takes bool variable, and calls SoccerBall constructor with "Arsenal", any price and the bool variable.	Total	MerchandiseFactory has pure virtual functions to create a SoccerBall Merchandise item and a Shirt Merchandise item.	ChelseaFactory implements functions to create a ChelseaSoccerBall Merchandise item and a ChelseaShirt, using their created constructors.	LiverpoolFactory implements functions to create a LiverpoolSoccerBall Merchandise item and a LiverpoolShirt, using their created constructors.	ArsenalFactory implements functions to create a ArsenalSoccerBall Merchandise item and a ArsenalShirt, using their created constructors.	Total	ChelseaFactory, LiverpoolFactory and ArsenalFactory are created.	Student asks user which products should be created, as well as options specific to that type of item.	Student creates concrete products using concrete creators, and outputs details of products created.	Total	Correct inheritance in the Merchandise hierarchy, between Merchandise and its child classes (SoccerBall and Shirt) and between these child classes and the concreteProducts.	Correct inheritance between concreteCreators and Creator (e.g. Factory and ChelseaFactory)	Correct associations between concreteCreators and concreteProducts (e.g. ChelseaFactory has associations to ChelseaSoccerBall and ChelseaShirt)	Total
u20581018	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u20592061	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	0	8
u20632429	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u20646284	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u20660652	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	0	2	5	5	3	6	14
u20662302	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14
u20692286	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	4	3	6	13
u20734621	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	0	5	5	3	6	14
u20780479	2	2	2	2	2	2	2	2	2	18	2	2	2	2	8	3	2	5	10	5	3	6	14

Practical Assignment 02

		Section B: State and Strategy Patterns																			
	TOTAL	SoccerPlayer Class	State Pattern							Strategy Pattern						Main				Object and State Diag	
	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6	
	TOTAL	SoccerPlayer class has a name parameter, as well as either getters and setters, or a constructor that sets the name.	Abstract CardState class has pure virtual handle() and changeCardState() functions, as well as a getter for the cardColour variable	NoCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of YellowCardState	YellowCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of RedCardState	RedCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of AddedCardState	AddedCardState private variable to SoccerPlayer class, as well as a commitFoul() function that calls the CardState's handle() and changeCardState() functions.	Total	Abstract PlayStyle has pure virtual play() function.	AttackPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	DefendPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	PossessionPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	Added PlayStyle private variable to SoccerPlayer class, as well as a play() function that tests whether a player is on a red card by calling the CardState's getCardColour() function, and either outputs the appropriate red card statement or calls PlayStyle's	Total	Student created a SoccerPlayer with a given name and playstyle	Student calls the SoccerPlayer's commitFoul method to show how the state changes from NoCard to YellowCard to RedCard, and then repeats on this state	Student calls the SoccerPlayer's play() function to show how a SoccerPlayer outputs different values based on it's PlayStyle strategy. Student also shows how a SoccerPlayer's play() function outputs the appropriate string when a player is on a red card and play() is called.	Total	Correct object diagram with aggregation relationships between SoccerPlayer and its concrete PlayStyle and CardState objects, and having a name value as well. The concrete CardState class must have a valid colour value.	Correct state diagram with starting and ending points, as well as states for having not received a card, a decision between receiving a yellow card or a red card based on the severity of a foul, and receiving a red card if a yellow card was received in	
u04483716	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6	
u04515146	26	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	6	
u04534205	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
u15223893	50	2	2	3	3	3	2	13	0	3	3	3	3	12	2	4	4	10	0	4	
u15231748	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	2	2	6	4	6	
u16068344	48	2	2	3	3	3	2	13	1	3	3	3	1	11	2	3	4	9	1	3	
u16320965	26	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	1	6	
u17016534	50	2	0	0	0	0	0	0	1	3	3	3	3	13	2	4	4	10	3	6	
u17029377	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	5	6	
u17030553	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
u17053928	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
u17066736	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	2	
u17102210	33	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	3	0	
u17110310	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	4	
u17169811	47	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	0	
u17229457	43	2	3	3	3	3	2	14	1	3	3	3	3	13	2	4	4	10	2	5	
u18019499	50	2	2	2	2	2	2	10	1	3	3	3	3	13	2	4	4	10	0	0	
u18025685	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	6	
u18034332	34	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0	
u18053239	50	2	2	3	3	3	2	13	0	3	3	3	3	12	2	4	4	10	4	5	
u18055215	40	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	0	
u18055461	50	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	
u18061908	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	3	
u18069704	32	2	2	3	3	3	2	13	1	3	3	3	3	13	2	0	2	4	0	0	
u18074074	36	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0	
u18080368	38	2	2	3	3	3	2	13	1	3	3	3	3	13	0	0	0	0	2	0	
u18105883	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
u18108467	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	4	
u18169253	42	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0	
u18191135	49	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	6	
u18196366	46	2	2	3	3	3	2	13	2	3	3	3	3	14	2	4	4	10	0	0	
u18219919	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0	
u18265163	42	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	4	
u18286250	48	2	2	3	3	3	2	13	0	3	3	3	3	12	2	4	4	10	3	4	
u18312374	47	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0	

Practical Assignment 02

		Section B: State and Strategy Patterns																		
	TOTAL	SoccerPlayer Class	State Pattern						Strategy Pattern						Main				Object and State Diag	
	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
	TOTAL	SoccerPlayer class has a name parameter, as well as either getters and setters, or a constructor that sets the name.	Abstract CardState class has pure virtual handle() and changeCardState() functions, as well as a getter for the cardColour variable	NoCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of YellowCardState	YellowCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of RedCardState	RedCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of AddedCardState	AddedCardState private variable to SoccerPlayer class, as well as a commitFoul() function that calls the CardState's handle() and changeCardState() functions.	Total	Abstract PlayStyle has pure virtual play() function.	AttackPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	DefendPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	PossessionPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	Added PlayStyle private variable to SoccerPlayer class, as well as a play() function that tests whether a player is on a red card by calling the CardState's getCardColour() function, and either outputs the appropriate red card statement or calls PlayStyle's	Total	Student created a SoccerPlayer with a given name and playstyle	Student calls the SoccerPlayer's commitFoul method to show how the state changes from NoCard to YellowCard to RedCard, and then repeats on this state	Student calls the SoccerPlayer's play() function to show how a SoccerPlayer outputs different values based on it's PlayStyle strategy. Student also shows how a SoccerPlayer's play() function outputs the appropriate string when a player is on a red card and play() is called.	Total	Correct object diagram with aggregation relationships between SoccerPlayer and its concrete PlayStyle and CardState objects, and having a name value as well. The concrete CardState class must have a valid colour value.	Correct state diagram with starting and ending points, as well as states for having not received a card, a decision between receiving a yellow card or a red card based on the severity of a foul, and receiving a red card if a yellow card was received in
u18335412	45	2	2	3	3	3	2	13	1	3	3	3	3	13	0	0	0	0	3	3
u18350110	46	2	3	3	3	3	2	14	1	3	3	3	2	12	2	4	4	10	0	0
u18371435	44	2	2	3	3	3	0	11	1	3	3	3	3	13	0	0	0	0	0	0
u19004232	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	6
u19009977	50	0	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	6
u19015951	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	6
u19021209	42	2	2	3	3	3	2	13	1	3	3	3	1	11	0	0	0	0	0	0
u19027372	26	2	2	3	3	2	2	12	1	3	3	3	0	10	2	4	4	10	6	6
u19028815	5	2	0	0	0	0	0	0	0	0	0	0	0	0	2	4	4	10	0	6
u19029242	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	6
u19033347	50	2	2	3	3	3	2	13	1	3	3	3	0	10	2	4	4	10	3	6
u19037717	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u19044233	44	2	2	3	3	3	2	13	1	3	3	3	0	10	0	0	0	0	0	0
u19048280	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	1	0
u19050993	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	6
u19053313	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	1	6
u19061359	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	5	6
u19062665	43	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	6
u19068035	43	2	2	3	3	2	0	10	1	3	3	3	3	13	2	0	0	2	5	6
u19068710	43	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	6
u19068752	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	6
u19072067	0							0						0				0		
u19083786	50	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u19100133	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u19103345	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	3	6
u19116498	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
u19123460	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
u19126353	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	3	0
u19130938	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u19141859	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u19151952	0							0						0				0		
u19153113	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	3	6
u19185032	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u19185678		2	2	3	3	2	2		1	3	3	3	3		2	4	4		2	6
u19198958	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	2	6

Practical Assignment 02

Section B: State and Strategy Patterns																				
	TOTAL	SoccerPlayer Class	State Pattern						Strategy Pattern						Main				Object and State Diag	
	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
	TOTAL	SoccerPlayer class has a name parameter, as well as either getters and setters, or a constructor that sets the name.	Abstract CardState class has pure virtual handle() and changeCardState() functions, as well as a getter for the cardColour variable	NoCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of YellowCardState	YellowCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of RedCardState	RedCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of AddedCardState	AddedCardState private variable to SoccerPlayer class, as well as a commitFoul() function that calls the CardState's handle() and changeCardState() functions.	Total	Abstract PlayStyle has pure virtual play() function.	AttackPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	DefendPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	PossessionPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	Added PlayStyle private variable to SoccerPlayer class, as well as a play() function that tests whether a player is on a red card by calling the CardState's getCardColour() function, and either outputs the appropriate red card statement or calls PlayStyle's	Total	Student created a SoccerPlayer with a given name and playstyle	Student calls the SoccerPlayer's commitFoul method to show how the state changes from NoCard to YellowCard to RedCard, and then repeats on this state	Student calls the SoccerPlayer's play() function to show how a SoccerPlayer outputs different values based on it's PlayStyle strategy. Student also shows how a SoccerPlayer's play() function outputs the appropriate string when a player is on a red card and play() is called.	Total	Correct object diagram with aggregation relationships between SoccerPlayer and its concrete PlayStyle and CardState objects, and having a name value as well. The concrete CardState class must have a valid colour value.	Correct state diagram with starting and ending points, as well as states for having not received a card, a decision between receiving a yellow card or a red card based on the severity of a foul, and receiving a red card if a yellow card was received in
u19214597	39	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u19228882	44	2	2	3	3	3	2	13	1	3	3	3	3	13		4	4	8	0	0
u19236183	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	6
u19236272	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u19290498	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	5	5
u19294418	36	2	2	3	3	3	2	13	1	3	3	3	4	14	3	4	4	11	0	0
u19367211	36	2	0	3	3	3	2	11	0	3	3	3	3	12	2	4	4	10	0	0
u20424622	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
u20426799	37	2	0	3	3	3	2	11	1	3	3	3	3	13	2	4	4	10	6	6
u20426918	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u20428082	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	3	3	8	5	6
u20430168	47	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u20430516	0							0						0				0		
u20430630	50	2	2	2	2	2	2	10	1	3	3	3	3	13	2	4	4	10	3	3
u20431997	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	6
u20432748	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	1	6
u20435780	36	2	0	3	3	3	2	11	1	3	3	3	3	13	2	4	4	10	0	0
u20435992	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	6
u20436077	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u20437863	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	4
u20438151	0							0						0				0		
u20439963	46	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	3
u20440562	50	2	2	3	3	3	2	13	1	3	3	3	3	13	0	0	0	0	0	0
u20441135	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u20442018	50	2	0	3	3	3	2	11	1	3	3	3	3	13	2	4	4	10	0	6
u20443260	36	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	4
u20443626	48	2	0	3	3	3	2	11	1	3	3	3	3	13	2	4	4	10	6	6
u20444738	49	2	0	3	3	3	2	11	1	3	3	3	3	13	2	3	3	8	0	6
u20448474	36	2	0	3	3	3	2	11	1	3	3	3	3	13	0	0	0	0	0	0
u20450533	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	1	5
u20450932	40	2	2	3	3	3	1	12	1	3	3	3	3	13	2	2	2	6	0	6
u20453222	48	2	2	3	3	3	2	13	1	3	3	3	3	13	2	2	3	7	1	6
u20453478	44	2	2	2	2	3	2	11	1	3	3	3	3	13	2	4	4	10	3	4
u20456078	36	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0
u20460067	50	2	2	2	2	2	2	10	1	3	3	3	3	13	2	4	4	10	3	5



		Section B: State and Strategy Patterns																			
		TOTAL	SoccerPlayer Class	State Pattern						Strategy Pattern						Main				Object and State Diag	
		50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
	TOTAL	SoccerPlayer class has a name parameter, as well as either getters and setters, or a constructor that sets the name.	Abstract CardState class has pure virtual handle() and changeCardState() functions, as well as a getter for the cardColour variable	NoCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of YellowCardState	YellowCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of RedCardState	RedCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of AddedCardState	AddedCardState private variable to SoccerPlayer class, as well as a commitFoul() function that calls the CardState's handle() and changeCardState() functions.	Total	Abstract PlayStyle has pure virtual play() function.	AttackPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	DefendPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	PossessionPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	Added PlayStyle's private variable to SoccerPlayer class, as well as a play() function that tests whether a player is on a red card by calling the CardState's getCardColour() function, and either outputs the appropriate red card statement or calls PlayStyle's	Total	Student created a SoccerPlayer with a given name and playstyle	Student calls the SoccerPlayer's commitFoul method to show how the state changes from NoCard to YellowCard to RedCard, and then repeats on this state.	Student calls the SoccerPlayer's play() function to show how a SoccerPlayer outputs different values based on it's PlayStyle strategy. Student also shows how a SoccerPlayer's play() function outputs the appropriate string when a player is on a red card and play() is called.	Total	Correct object diagram with aggregation relationships between SoccerPlayer and its concrete PlayStyle and CardState objects, and having a name value as well. The concrete CardState class must have a valid colour value.	Correct state diagram with starting and ending points, as well as states for having not received a card, a decision between receiving a yellow card or a red card based on the severity of a foul, and receiving a red card if a yellow card was received in	
u20460687	36	2	2	3	3	3	2	13	1	3	3	3	3	13	0	0	0	0	0	0	
u20463163	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	0	0	2	3	3	
u20465026	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	5	6	
u20466570	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
u20468203	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	0	0	
u20469366	44	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	3	6	
u20471582	40	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	3	1	
u20478144	16	2	2	3	3	2	2	12	1	0	0	0	3	4	0	0	0	0	0	0	
u20479884	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	1	6	
u20481218	40	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	2	8	1	6	
u20486783	29	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	3	3	
u20491141	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	3	6	
u20494166	50	2	2	3	3	3	2	13	1	3	3	3	4	14	3	4	4	11	6	0	
u20494654	48	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	3	6	
u20498510	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	4	
u20502126	25	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	4	4	
u20504552	49	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	1	3	
u20506237	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	4	6	
u20507102	41	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	4	2	
u20513667	31	2	2	3	3	2	2	12	1	3	3	3	3	13	0	0	0	0	4	4	
u20519517	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	0	0	
u20522623	36	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	0	0	
u20528036	50	2	2	3	3	2	2	12	1	3	3	3	3	13	2	3	4	9	2	4</	

Practical Assignment 02

Section B: State and Strategy Patterns																				
	TOTAL	SoccerPlayer Class	State Pattern						Strategy Pattern						Main				Object and State Diag	
	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
	TOTAL	SoccerPlayer class has a name parameter, as well as either getters and setters, or a constructor that sets the name.	Abstract CardState class has pure virtual handle() and changeCardState() functions, as well as a getter for the cardColour variable	NoCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of YellowCardState	YellowCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of RedCardState	RedCardState class has a constructor that sets the cardColour variable, as well as a handle function that outputs the appropriate sentence and a changeCardState that returns a new instance of AddedCardState	AddedCardState private variable to SoccerPlayer class, as well as a commitFoul() function that calls the CardState's handle() and changeCardState() functions.	Total	Abstract PlayStyle has pure virtual play() function.	AttackPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	DefendPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	PossessionPlayStyle inherits from PlayStyle and implements PlayStyle's play() function, which prints the appropriate string	Added PlayStyle private variable to SoccerPlayer class, as well as a play() function that tests whether a player is on a red card by calling the CardState's getCardColour() function, and either outputs the appropriate red card statement or calls PlayStyle's	Total	Student created a SoccerPlayer with a given name and playstyle	Student calls the SoccerPlayer's commitFoul method to show how the state changes from NoCard to YellowCard to RedCard, and then repeats on this state.	Student calls the SoccerPlayer's play() function to show how a SoccerPlayer outputs different values based on it's PlayStyle strategy. Student also shows how a SoccerPlayer's play() function outputs the appropriate string when a player is on a red card and play() is called.	Total	Correct object diagram with aggregation relationships between SoccerPlayer and its concrete PlayStyle and CardState objects, and having a name value as well. The concrete CardState class must have a valid colour value.	Correct state diagram with starting and ending points, as well as states for having not received a card, a decision between receiving a yellow card or a red card based on the severity of a foul, and receiving a red card if a yellow card was received in
u20581018	50	2	2	3	3	3	2	12	1	3	3	3	3	13	2	4	4	10	2	2
u20592061	44	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	2	4
u20632429	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
u20646284	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
u20660652	45	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	3	6
u20662302	50	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	5	6
u20692286	49	2	2	3	3	3	2	13	1	3	3	3	3	13	2	4	4	10	6	6
u20734621	45	2	2	3	3	2	2	12	1	3	3	3	3	13	2	4	4	10	4	5
u20780479	50	2	2	3	3		2	12	1	3	3	3	3	13	2	4	2	8	0	0

Practical Assignment 02

	rams		Final Mark
	12	50	100
	Total	TOTAL	FINAL MARK
u04483716	12	50	94
u04515146	6	44	70
u04534205	0	0	11
u15223893	4	41	91
u15231748	10	44	92
u16068344	4	39	87
u16320965	7	45	71
u17016534	9	34	84
u17029377	11	49	93
u17030553	0	0	0
u17053928	0	0	50
u17066736	6	44	94
u17102210	3	41	74
u17110310	4	42	86
u17169811	4	42	89
u17229457	7	46	89
u18019499	0	35	85
u18025685	8	46	90
u18034332	0	38	72
u18053239	9	46	96
u18055215	4	42	82
u18055461	0	2	52
u18061908	7	45	93
u18069704	0	32	64
u18074074	0	38	74
u18080368	2	30	68
u18105883	0	0	50
u18108467	6	44	88
u18169253	0	38	80
u18191135	8	46	95
u18196366	0	39	85
u18219919	0	38	88
u18265163	6	44	86
u18286250	7	44	92
u18312374	0	38	85

Practical Assignment 02

	rams		Final Mark
	12	50	100
	Total	TOTAL	FINAL MARK
u18335412	6	34	79
u18350110	0	38	84
u18371435	0	26	70
u19004232	6	44	92
u19009977	10	46	96
u19015951	6	44	92
u19021209	0	26	68
u19027372	12	46	72
u19028815	6	18	23
u19029242	6	44	94
u19033347	9	44	94
u19037717	0	38	86
u19044233	0	25	69
u19048280	1	39	83
u19050993	8	8	50
u19053313	7	45	95
u19061359	11	49	99
u19062665	10	48	91
u19068035	11	38	81
u19068710	10	48	91
u19068752	6	44	94
u19072067	0	0	0
u19083786	0	2	52
u19100133	0	0	18
u19103345	9	47	97
u19116498	12	50	94
u19123460	12	50	100
u19126353	3	41	91
u19130938	0	38	88
u19141859	0	38	88
u19151952	0	0	0
u19153113	9	47	97
u19185032	0	0	50
u19185678	8	10	10
u19198958	8	45	95

Practical Assignment 02

	rams		Final Mark
	12	50	100
	Total	TOTAL	FINAL MARK
u19214597	0	38	77
u19228882	0	36	80
u19236183	6	44	94
u19236272	0	0	6
u19290498	10	48	96
u19294418	0	40	76
u19367211	0	35	71
u20424622	12	50	98
u20426799	12	48	85
u20426918	0	38	88
u20428082	11	47	97
u20430168	0	38	85
u20430516	0	0	0
u20430630	6	41	91
u20431997	8	46	96
u20432748	7	45	95
u20435780	0	36	72
u20435992	8	46	96
u20436077	0	0	2
u20437863	6	44	92
u20438151	0	0	0
u20439963	5	43	89
u20440562	0	28	78
u20441135	0	0	50
u20442018	6	42	92
u20443260	8	46	82
u20443626	12	48	96
u20444738	6	40	89
u20448474	0	26	62
u20450533	6	44	92
u20450932	6	39	79
u20453222	7	42	90
u20453478	7	43	87
u20456078	0	38	74
u20460067	8	43	93

Practical Assignment 02

	rams		Final Mark
	12	50	100
	Total	TOTAL	FINAL MARK
u20460687	0	28	64
u20463163	6	36	86
u20465026	11	48	98
u20466570	0	0	24
u20468203	0	37	87
u20469366	9	46	90
u20471582	4	41	81
u20478144	0	18	34
u20479884	7	44	94
u20481218	7	42	82
u20486783	6	43	72
u20491141	9	46	96
u20494166	6	46	96
u20494654	9	46	94
u20498510	8	46	96
u20502126	8	46	71
u20504552	4	41	90
u20506237	10	47	97
u20507102	6	43	84
u20513667	8	35	66
u20519517	0	37	87
u20522623	0	38	74
u20528036	6	42	92
u20528834	6	33	73
u20529440	0	27	66
u20532581	3	39	79.5
u20534541	10	47	96
u20536951	7	43	87
u20538945	1	28	67
u20554240	6	33	72
u20556455	8	45	95
u20557622	9	46	96
u20573783	0	27	69
u20575085	5	42	92
u20578688	0	0	44

Practical Assignment 02

		rams	Final Mark
		12	50
		100	
		Total	TOTAL
			FINAL MARK
u20581018	4	41	91
u20592061	6	44	88
u20632429	12	50	100
u20646284	12	50	100
u20660652	9	47	92
u20662302	11	49	99
u20692286	12	50	99
u20734621	9	46	91
u20780479	0	35	85