CSC2018 Advisory: Getting Started with Eclipse

Create a new Project in Eclipse

- Select *File...New...Java Project*
- Let's call the project *Student*. Select the defaults. (You may wish to select the latest *Java Runtime Environment* (JRE), which at time of writing is JavaSE-1.7). Click *Finish*.
- Look in the *Package Explorer* – if it isn't open already select (*Window...Show View...Package Explorer*). By default the *Package Explorer* appears as a column to the left of the Eclipse window. You'll see that a new project *Student* has appeared.
- Click the white arrowhead beside *Student* to see what Eclipse has created for you. You'll see that one of the new folders created is the source folder *src*.
- It's often useful to organise code into packages. Right-click *src* and select *New...Java Package*. You need to enter the name of the new package in the *Name* field: we'll call the package *studentPackage*. Remember to adhere to the conventions for use of upper and lower case in Java!

Putting some code into the package

In this section we'll create some classes to put in our package. The aim here is not only to use Eclipse, but to gain some experience in good object-oriented coding: creating classes that inherit from each other so that common features need appear only once. The attributes and methods of each class are described in this worksheet, as are the relationships between the classes. Try to do as much as you can by yourself. If you have difficulty, check the listings at the back of this worksheet. Use ctrl + s *to save each of your new classes as you create them. Try to stick as closely as you can to the specifications outlined, but if your suite of classes is a little different from the ones described here, that's still OK. The important thing is that you get some experience of using Eclipse and the facilities it provides you for creating good object-oriented code.*

We're going to create a very simple 'community' of objects representing students of different kinds. Naturally enough, students are also people that have a name, so we'll start off by creating a *Name* class, which in turn will be an attribute of class *Person*.

Name

- Right-click *studentPackage* and select *New...Class*. Name the class *Name*. Uncheck *public static void main(String[] args)* (as this will not be the not be 'main' class *that drives the program – we'll add that at the end of our exercise). In the Package* Explorer you'll see a new class *Name.java* appear under *studentPackage*.
- Double-click *Name.java*. *Name* is going to consist of a *first name* and a *last name* or a *first name,* a *last name* and a *middle name*, so we'll need constructors that can make *Name* objects with these different sets of attributes, as well as methods to set and get the *first*, *last* and *middle name* strings. Also include a method that will return the full name as a single *String*. Enter the new code that you need between the braces ({...}) that Eclipse has provided.
- Use *ctrl + s* to save your new class.

Person

- Create a new class in *StudentPackage* as before. Name the class *Person*.
- *Person* will have the private attributes *name* (of type *Name*), *sex* (a *char* – 'm' or 'f') and *id* (a *String*). Create constructors for *Person* that can be used both with and without *id.*
- Add methods that can get the *name*, *sex* and *id* of the person, and a method that can set the *id* of the person.
- Add a method *toString( )* that can return all the person's details in the form of a *String*.
- Use *ctrl + s* to save your new class.

Student

- Create a new class in *StudentPackage* as before. Name the class *Student*. Student will be a subclass of *Person*, so you'll need to add *extends Person* just before the braces ({..}) that Eclipse provides you.
- In addition to the attributes of *Student*, *Person* and will have attributes *credits* (an *int*) and *average_mark* (a *double*).
- Write methods that return the values for *credits* and *average mark*, as well as a method that can return a full report in the form of a *String* that includes all the student's details. There is an opportunity here to create a method that makes use of the value returned by the *toString( )* method of the superclass (*super.toString( )*).
- Include the following method, that increases the student's credits depending on the mark that they have obtained and the type of 'module part' (half, full or double) that they have taken.

```
public void update(double smark, String modulept)
{   int increase=0;
    modulept=modulept.toLowerCase();
    if (smark >= 40){
      if (modulept.equals("half_module"))
        increase= 10;
      else
         if (modulept.equals("full_module"))
            increase= 20;
         else
            if (modulept.equals("double_module"))
               increase= 40;
            else
               System.out.println("Module type not recognised - no credits added");
    }//if
    credits += increase;
}//update
```

CollegeStudent

- Create a new class *CollegeStudent* that inherits from *Student*. This has a new attribute *year*. *CollegeStudent*'s constructor can make use of its superclass's constructor.
- Give *CollegeStudent* a method that can return the *year* attribute and another that can return all of a CollegeStudent's details as a *String*.

GradStudent

- The new class *GradStudent* inherits from *CollegeStudent* and has a new attribute *degree* of type *String*.
- Write methods as before that let you access *degree* and also incorporate it in a detailed *GradStudent* report of type *String*.

TryAllStudents

- Now create a class that includes a *main* method (tick the checkbox on the window that you use to create the class) that will allow you to put your family of students through their paces. Let's call this new class *TryAllStudents.java*.
- Start off by creating a new *GradStudent* and returning his/her details.
- To compile and run your program, right-click your *Student* project in the *Package Explorer* and select *Run As...Java Application*. If everything is working properly, the output will appear in a *Console* window at the bottom of your screen.

Code listings

Name.java

```
package studentPackage;

public class Name {
	private String first;
	private String middle;
	private String last;

	public Name()
	{ //default constructor
	}

	public Name(String firstn, String lastn)
	{	first = firstn;
		last = lastn;
	}

	public Name(String firstn, String middlen, String lastn)
	{	first = firstn;
		middle = middlen;
		last = lastn;
	}

	public String getFirst()
	{ return first;
	}

	public String getMiddle()
	{ return middle;
	}

	public String getLast()
	{ return last;
	}

	public void setFirst(String firstn)
	{first = firstn;
	}

	public void setMiddle(String firstn)
	{first = firstn;
	}

	public void setLast(String firstn)
	{first = firstn;
	}

	public String toString()
	{ String s = new String();
		if (first != null) s = s + first + " ";
		if (middle != null) s = s + middle + " ";
		if (last != null) s = s + last;
		return s;
	}
}
```

Person.java

```
package studentPackage;

public class Person {
	protected Name name;
	protected char sex;
	protected String id;

	public Person (Name pname, char psex)
	{	name = pname;
		sex = psex;
	}

	public Person (Name pname, char psex, String pid)
	{	name = pname;
		sex = psex;
		id = pid;
	}

	public Name getname()
	{  return name;
	}

	public char getsex()
	{  return sex;
	}


	public String getid()
	{  return id;
	}

	public void setid(String sid)
	{ id=sid;
	}

	public String toString()
	{ String s = new String("\n" + name + "\n" + "\tsex: " + sex);
	  if (id !=null) s = s + " id: " + id;
	  return s;
	}

}
```

Student.java

```
package studentPackage;

public class Student extends Person {
	protected int credits;
	protected double avg_mark;

	public Student(Name name, char sex, int credits, double mark)	{
		super(name, sex);
		this.credits=credits;
		this.avg_mark = mark;
	}//Student

	public int credits()
	{ return credits;
	}//credits

	double avg_mark()
	{  return avg_mark();
	}//avg_mark

	public String toString()
	{
		String s;
		s= new String(super.toString());
		s = s + "\n\tcredits: " + credits;
		s = s + "\n\tmark average: " + avg_mark;
		return s;
	}//toString

	public void update(double smark, String modulept)
	{	int increase=0;

		modulept=modulept.toLowerCase();
		if (smark >= 40)
		{  if (modulept.equals("half_module"))
			increase= 10;
		   else
		      if (modulept.equals("full_module"))
		         increase= 20;
		      else
		         if (modulept.equals("double_module"))
		            increase= 40;
		         else
		            System.out.println("Module type not recognised
		                                              - no credits added");
		}//if
		credits += increase;
	}//update

}
```

CollegeStudent.java

```
package studentPackage;

public class CollegeStudent extends Student {

	protected String year;

	public CollegeStudent(Name sname, char ssex, int scredits,
                                                    double smark, String syear)
	{ super(sname, ssex, scredits, smark);
		year = syear;
	}

	public String getyear()
	{ return year;
	}

	public String toString()
	{ String s = new String(super.toString());
		s = s + "\n\tyear: " + year;
		return s;
	}
}
```

GradStudent.java

```
package studentPackage;

public class GradStudent extends CollegeStudent {

	protected String degree;

	public GradStudent(Name sname, char ssex, int scredits,
                                    double smark, String syear, String sdegree)
	{
		super(sname, ssex, scredits, smark, syear);
		degree = sdegree;
	}

	public String degree()
	{ return degree;
	}

	public String toString()
	{ String s = new String(super.toString());
	s+= "\n\tdegree: " + degree;
	return s;
	}
}
```

TryAllStudents.java

```
package studentPackage;

public class TryAllStudents {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		Name name1 = new Name("Bill", "Smith");
		GradStudent student1 =
				new GradStudent(name1, 'M', 100, 45, "1998", "Physics");
		System.out.println("Grad student = " + student1);
	}
}
```