Abstraction

Abstraction is the strategic removal of detail from an algorithm

Exercise 1
Which algorithm is more abstract and why?

(a) i)

```
Algorithm ReplaceToiletPaper:

find new toilet paper roll
remove empty roll from paper roll holder
put new toilet paper roll onto paper roll holder
toss empty roll container
```

ii)

```
Algorithm ReplaceToiletPaper:

find new toilet paper roll
replace empty roll with new toilet paper roll
toss empty roll container
```

Answer: ii) is more abstract.

(b) i)

```
Algorithm DealCards:

for each player in the game:
     deal the player five cards
```

ii)

```
Algorithm DealCards:

for each player in the game:
     while player does not have five cards in hand:
          draw a card from the top of the deck
          give the player the drawn card
```

Answer: i) is more abstract.

(c) i)

```
Algorithm MakeSnowman:

roll three balls of snow
stack balls of snow
put sticks in middle ball for arms
```

ii)

```
Algorithm MakeSnowman:

roll large ball of snow
roll medium ball of snow
roll small ball of snow
put medium ball of snow on top of large ball of snow
put small ball of snow on top of medium ball of snow
put sticks in middle ball for arms
```

Answer: i) is more abstract.

Exercise 1

As with most other abstract exercises (pun intended), answers may vary wildly. As long as relevant details are removed/expanded based on whether the question asks for an abstraction or refinement, then the resulting algorithm should be considered "correct". Code modifications are highlighted in red.

(a) Write a more abstract version of this algorithm:

```
Algorithm FireCannon:

bring cannonball to cannon
drop cannonball into cannon
pack cannonball tightly into cannon
run to back of cannon
find and grab cannon's rope trigger
pull cannon rope trigger
```

Solution

```
Algorithm FireCannon:

load cannonball into cannon
pull rope trigger
```

(b) Refine the instruction `brush teeth`:

```
Algorithm PrepareForBed:

change into pajamas
brush teeth
floss teeth
```

Solution

```
Algorithm PrepareForBed:

change into pajamas
squeeze toothpaste from toothpaste tube onto toothbrush
rinse teeth with water
brush toothbrush against teeth and rinse a few times
rinse toothbrush and put it aside
floss teeth
```

(c) Write both an abstraction AND a refinement of this algorithm (two separate answers):

```
Algorithm MailItem:

pack item in a box
apply postage information onto box
find nearest mailbox
drop box in mailbox
```

Solution

i) Abstraction (here we abstracted all lines)

```
Algorithm MailItem:

pack item for delivery
deliver packaged item to mailbox
```

ii) Refinement (here we refined the first two lines only- any/every line can be refined)

```
Algorithm MailItem:

find box that item will fit in
place item into box
pad item in box with bubble wrap
write sender information onto paper
write recipient information onto paper
secure paper postage information onto box
find nearest mailbox
drop box in mailbox
```

Exercise 2

Perform encapsulations on the following algorithms:

As with most other abstract exercises (pun intended), answers may vary wildly. As long as relevant encapsulations are performed, then the resulting solution should be okay. Code modifications are highlighted in red.

(a) Perform at least one encapsulation.

```
Algorithm CallEmergencyNumber:

find nearest phone
pick up phone receiver
```

```
press ’9’ on dial pad
press ’1’ on dial pad
press ’1’ on dial pad
wait for emergency operator to pick up
```

Solution

```
Algorithm Dial911:

press ’9’ on dial pad
press ’1’ on dial pad
press ’1’ on dial pad
```

Solution

```
Algorithm CallEmergencyNumber:

find nearest phone
pick up phone receiver
Dial911
wait for emergency operator to pick up
```

(b) Perform at least two encapsulations.

```
Algorithm CalculateSalesTotal:

determine total cost of all items
apply manufacturer’s coupons
apply store coupons
add provincial tax
add federal tax
report final total
```

Solution

```
Algorithm ApplyCoupons:

apply manufacturer’s coupons
apply store coupons
```

Solution

```
Algorithm ApplyTaxes:

add provincial tax
add federal tax
```

Solution

```
Algorithm CalculateSalesTotal:

determine total cost of all items
ApplyCoupons
ApplyTaxes
report final total
```

Extra Exercises

Exercise 2 (ctn'd)

(c) Perform at least two encapsulations.

```
Algorithm FireCannon:

bring cannonball to cannon
drop cannonball into cannon
pack cannonball tightly into cannon
run to back of cannon
find and grab cannon's rope trigger
pull cannon rope trigger to fire cannonball
```

Solution

```
Algorithm LoadCannon:

bring cannonball to cannon
drop cannonball into cannon
pack cannonball tightly into cannon
```

Solution

```
Algorithm PullTrigger:

run to back of cannon
find and grab cannon's rope trigger
pull cannon rope trigger to fire cannonball
```

Solution

```
Algorithm FireCannon:

LoadCannon
PullTrigger
```