Question 2.4

<rainRecord>

<rain date="2023/01/01">

<station>AagA100</station>

<mm>10</mm>

</rain>

<rain date="2023/01/02">

<station>BcdB123</station>

<mm>5</mm>

</rain>

</rainRecord>

QString RainXml::writeToXml(/\*passing rain data\*/)

{

QString xmlOutput;

QXmlStreamWriter writer(&xmlOutput);

// do initial setup of xml text

writer.writeStartDocument();

writer.writeStartElement(“rainRecord”)

// loop through each rain pointer named r (do not code this)

{

const QMetaObject \*meta = r->metaObject();

QString data = (meta->property(meta->

indexOfProperty("data"))).read(r).toString();

// alternative

QString data = (meta->property(1)).read(r).toString();

QStringList list(data().split(":"));

// use the meta-object to get the required data

//if the station code passes the test

if (checkStationCode(list.at(0)))

{

// set up the <rain> tag and its sub-tags as required

writer.writeStartElement("rain");

QXmlStreamAttribute attr("date", list.at(1));

writer.writeAttribute(attr);

writer.writeTextElement("station", list.at(0));

writer.writeTextElement("mm", list.at(2));

writer.writeEndElement();

}

}

// end xml text

writer.writeEndElement();

writer.writeEndDocument();

return xmlOutput;

}

Question 3.1

class StationThread : public QObject

{

Q\_OBJECT

public:

StationThread(/\*all data\*/, QString stn);

public slots:

void doSearch();

signals:

void foundStation(QString date, QString mm);

private:

/\*all data type\*/ \*record;

QString station;

};

Question 3.2

QThread \*t{new QThread};

StationThread \*st{new StationThread(/\*passing parameters\*/)};

st->moveToThread(t);

connect(t, SIGNAL(started()), st, SLOT(doSearch()));

connect(st, SIGNAL(foundStation(QString,QString)), this,

SLOT(handleFound(QString,QString)));

// alternative

connect(st, &StationThread::foundStation, this,

&MainWindow::handleFound);

t->start();