Activity No. 13 Geolocation using Expo

Course Code: CPE026	**Program:** Computer Engineering
Course Title: Emerging Technologies 3 in CpE	**Date Performed:** November 16, 2024
Section: CPE41S8	**Date Submitted:** November 30, 2024
Name: Alferos, Joshua L. Efa, Christian Ed B.	**Instructor:** Engr. Roman Richard

1. Objectives

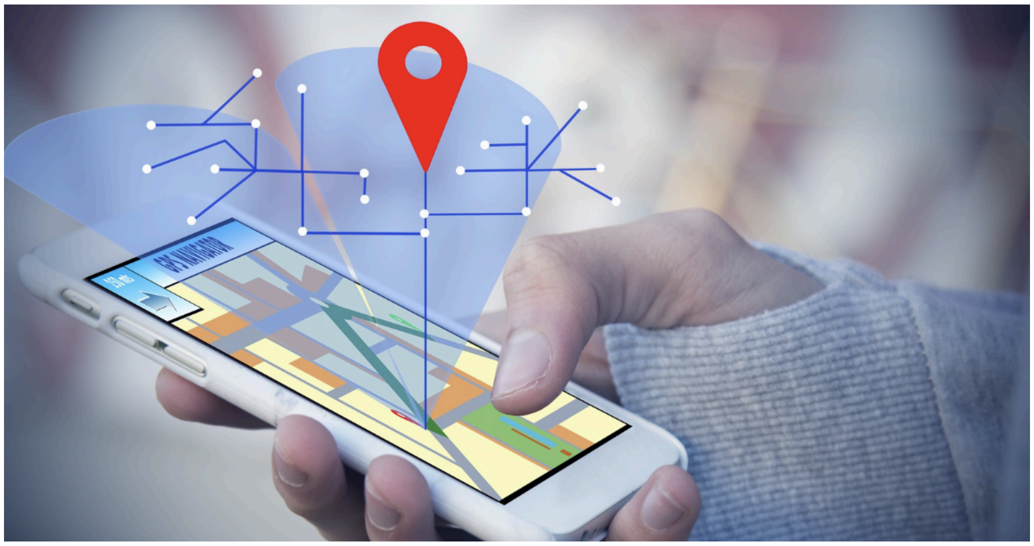
This activity aims to introduce students to the use and implementation of Geolocation on a Mobile Application built in React Native with Expo.

2. Intended Learning Outcome

After this module, the students should be able to:

- Demonstrate the use of geolocation on a mobile application built in React Native through Expo location.

3. Discussion



The React Native geolocation API is slightly different from other APIs: we can access it directly from the global navigator object, rather than importing it at the top of the file.

The geolocation API in React Native is the same as the one found in modern web browsers. This means better compatibility between libraries and a lower learning curve if you're coming from web development. On the web, the navigator object contains a lot of useful metadata about your web browser. In React Native, it's really just a container for geolocation and potentially a handful of other browser APIs. Accessing a global variable feels a bit unusual in React Native, but is necessary to provide the exact same API on web and mobile.

We'll use navigator.geolocation.getCurrentPosition to get our current position. This API takes a callback parameter which is called with an object containing our coordinates, coords, in latitude and longitude.

4. Materials and Equipment

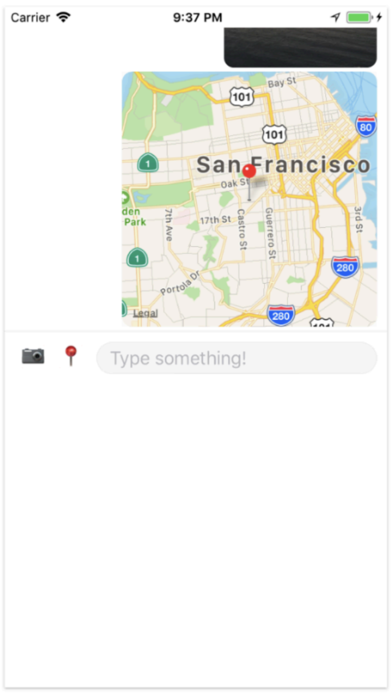
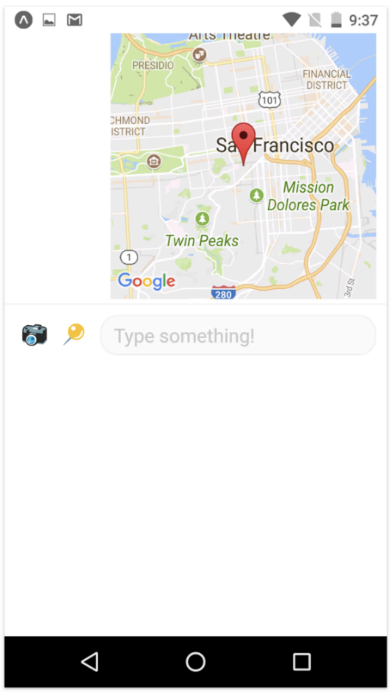
- Nodejs LTS
- Visual Studio Code
- Emulator/Simular for Android/iOS

5. Procedure

Let’s try it out. We can get our current position and use it to create a location message in the MessageList. Add the following to handlePressToolbarLocation in App.js:

```
//...
handlePressToolbarLocation = () => {
  const { messages } = this.state;
  navigator.geolocation.getCurrentPosition((position) => {
    const {
      coords: { latitude, longitude },
    } = position;
    this.setState({
      messages: [
        createLocationMessage({
          latitude,
          longitude,
        }),
        ...messages,
      ],
    });
  });
};
// ...
```

Pretty simple! If you try it out, you may be prompted to give Expo permission to access your location. Expo is already set up to allow the location permission. If you’re building an app using react-native-cli, you’ll also need to modify your Info.plist on iOS and AndroidManifest.xml on Android to enable location permissions. Tapping the location button should now add a location message:

iOS	Android
	

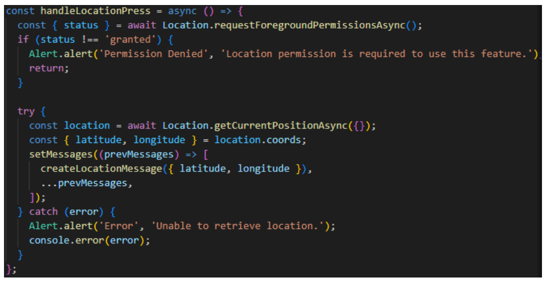
Did you also get the same result? Show screenshots for this.

Depending on how we're using geolocation, there are a few other APIs that might be useful: - watchPosition(success, error?, options?) and clearWatch(watchID) can be used to receive notifications when location changes. We can also pass the options timeout (number in ms), maximumAge (number in ms), and enableHighAccuracy (bool) for more granular control. - requestAuthorization() can be used to request access to device location. This can be a better experience than presenting an alert when a map is shown for the first time. - getCurrentPosition(geo_- success, geo_error?, geo_options?) is the full function signature of the getCurrentPosition API we use above. Although we didn't do it in our example, we would generally want to handle errors and present them to the user in some way. We might also want to pass options for more granular control (the same options as watchPosition).

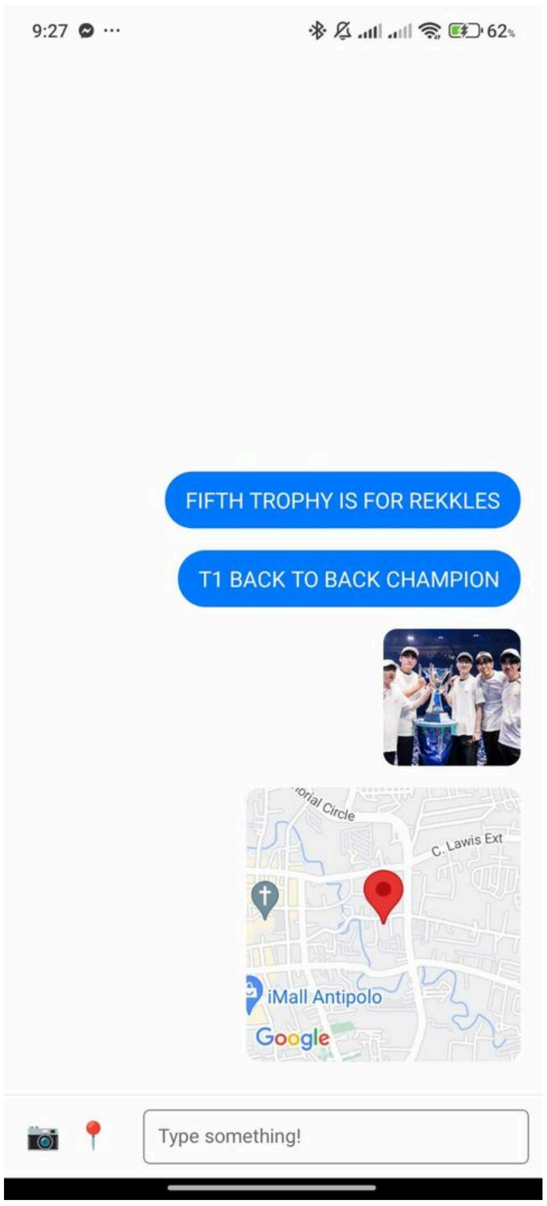
7. Output

ALFEROS

Code	**Output**



```
const handleLocationPress = async () => {
  const { status } = await Location.requestForegroundPermissionsAsync();
  if (status !== 'granted') {
    Alert.alert('Permission Denied', 'Location permission is required to use this feature.');
    return;
  }

  try {
    const location = await Location.getCurrentPositionAsync({});
    const { latitude, longitude } = location.coords;
    setMessages((prevMessages) => [
      createLocationMessage({ latitude, longitude }),
      ...prevMessages,
    ]);
  } catch (error) {
    Alert.alert('Error', 'Unable to retrieve location.');
    console.error(error);
  }
};
```



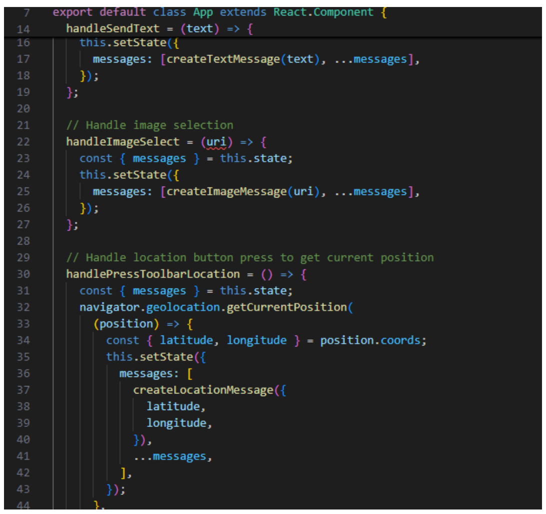
FIFTH TROPHY IS FOR REKKLES

T1 BACK TO BACK CHAMPION

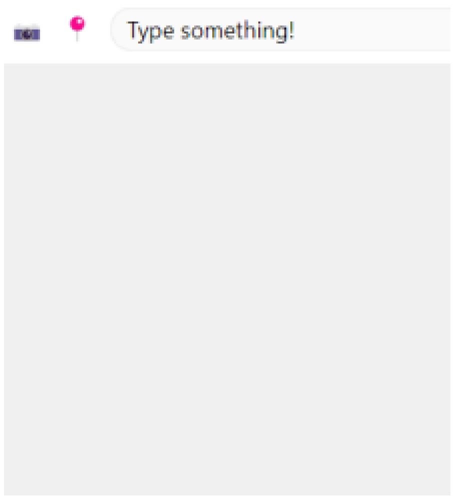
Did you also get the same result?

Yes I got the same result.

EFA



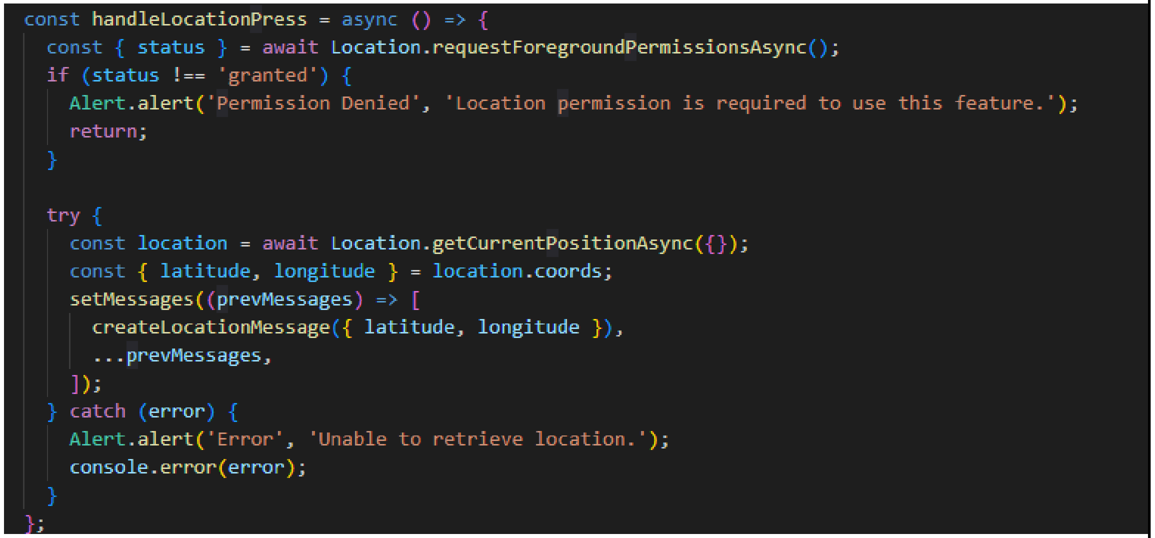
```
export default class App extends React.Component {
  handleSendText = (text) => {
    this.setState({
      messages: [createTextMessage(text), ...messages],
    });
  };

  // Handle image selection
  handleImageSelect = (uri) => {
    const { messages } = this.state;
    this.setState({
      messages: [createImageMessage(uri), ...messages],
    });
  };

  // Handle location button press to get current position
  handlePressToolbarLocation = () => {
    const { messages } = this.state;
    navigator.geolocation.getCurrentPosition(
      (position) => {
        const { latitude, longitude } = position.coords;
        this.setState({
          messages: [
            createLocationMessage({
              latitude,
              longitude,
            }),
            ...messages,
          ],
        });
      },
```



8. Supplementary Activity

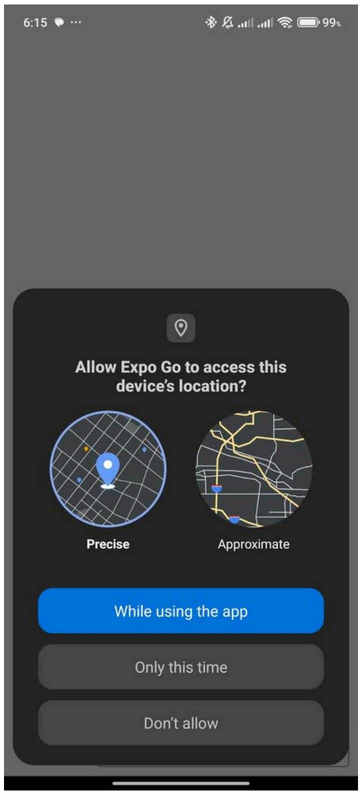
1. Include geolocation in your application. Screenshot the output.
2. Demonstrate that the geolocation feature is working in the messaging application.

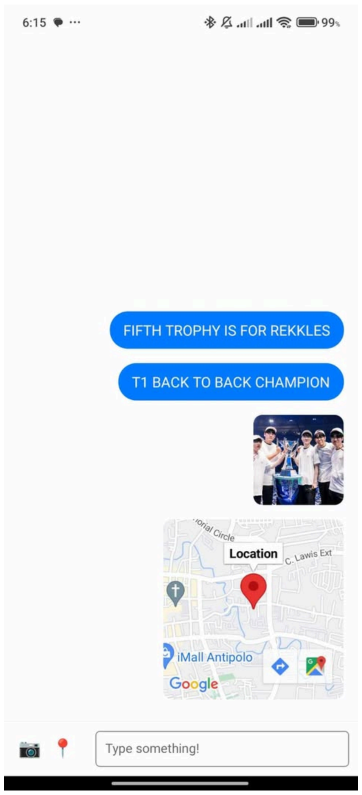
Code



```
const handleLocationPress = async () => {
  const { status } = await Location.requestForegroundPermissionsAsync();
  if (status !== 'granted') {
    Alert.alert('Permission Denied', 'Location permission is required to use this feature.');
    return;
  }

  try {
    const location = await Location.getCurrentPositionAsync({});
    const { latitude, longitude } = location.coords;
    setMessages((prevMessages) => [
      createLocationMessage({ latitude, longitude }),
      ...prevMessages,
    ]);
  } catch (error) {
    Alert.alert('Error', 'Unable to retrieve location.');
    console.error(error);
  }
};
```

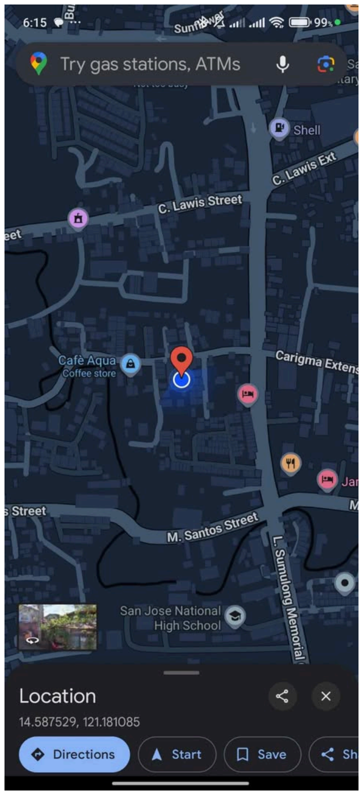
Demonstration





FIFTH TROPHY IS FOR REKKLES

T1 BACK TO BACK CHAMPION



Location
14.587529, 121.181085

9. Conclusion

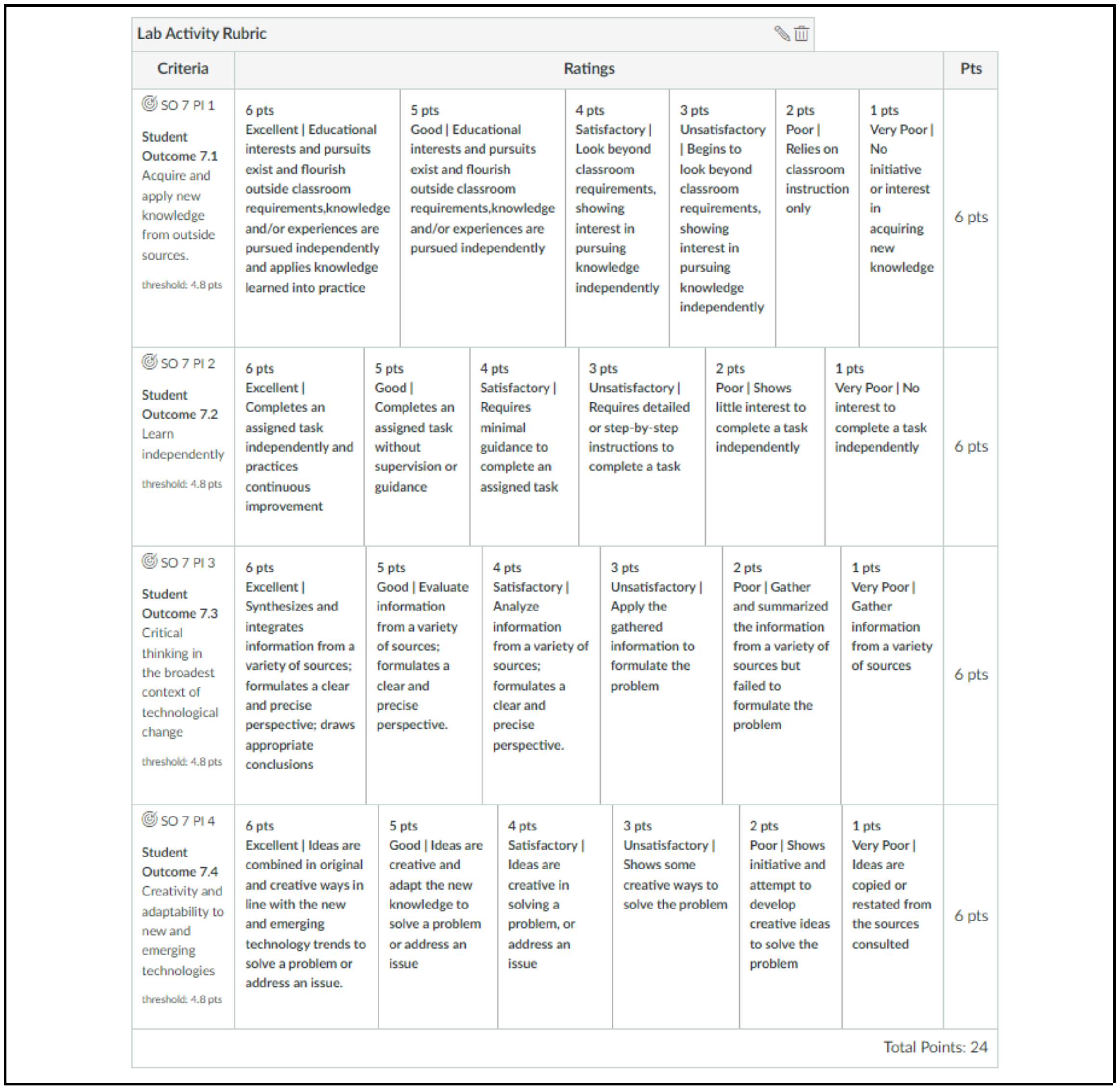
ALFEROS

In this activity, we explored the use and implementation of Geolocation using React Natie in Expo. Geolocation can be easily implemented using the expo API, which enables the the user to access and send their location data. The geolocation offers built in methods for retrieving user's current location, and monitor changes in location. Overall, this activity showed us practical application of geolocation in mobile application development

EFA

-I conclude that I have learned that utilizing geolocation in a React Native application can be easily achieved using the expo-location API, which simplifies the process of accessing and managing location data. With proper permissions and error handling, developers can integrate location features into their apps, improving the user experience through location-based functionality on both mobile and web platforms.

10, Assessment Rubric



Lab Activity Rubric

Criteria	Ratings						Pts
SO 7 PI 1 **Student Outcome 7.1** Acquire and apply new knowledge from outside sources. threshold: 4.8 pts	6 pts Excellent | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently	4 pts Satisfactory | Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory | Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor | Relies on classroom instruction only	1 pts Very Poor | No initiative or interest in acquiring new knowledge	6 pts
SO 7 PI 2 **Student Outcome 7.2** Learn independently threshold: 4.8 pts	6 pts Excellent | Completes an assigned task independently and practices continuous improvement	5 pts Good | Completes an assigned task without supervision or guidance	4 pts Satisfactory | Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory | Requires detailed or step-by-step instructions to complete a task	2 pts Poor | Shows little interest to complete a task independently	1 pts Very Poor | No interest to complete a task independently	6 pts
SO 7 PI 3 **Student Outcome 7.3** Critical thinking in the broadest context of technological change threshold: 4.8 pts	6 pts Excellent | Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good | Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory | Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory | Apply the gathered information to formulate the problem	2 pts Poor | Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor | Gather information from a variety of sources	6 pts
SO 7 PI 4 **Student Outcome 7.4** Creativity and adaptability to new and emerging technologies threshold: 4.8 pts	6 pts Excellent | Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good | Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory | Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory | Shows some creative ways to solve the problem	2 pts Poor | Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor | Ideas are copied or restated from the sources consulted	6 pts
							Total Points: 24